

**NASA
Technical
Memorandum**

NASA TM - 108409

**COMPUTERIZED ATMOSPHERIC TRACE CONTAMINANT
CONTROL SIMULATION FOR MANNED SPACECRAFT**

By J.L. Perry

**Structures and Dynamics Laboratory
Science and Engineering Directorate**

June 1993

(NASA-TM-108409) COMPUTERIZED
ATMOSPHERIC TRACE CONTAMINANT
CONTROL SIMULATION FOR MANNED
SPACECRAFT (NASA) 171 p

N93-28977

Unclass

G3/54 0174962



National Aeronautics and
Space Administration

George C. Marshall Space Flight Center

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE June 1993	3. REPORT TYPE AND DATES COVERED Technical Memorandum		
4. TITLE AND SUBTITLE Computerized Atmospheric Trace Contaminant Control Simulation for Manned Spacecraft			5. FUNDING NUMBERS	
6. AUTHOR(S) J.L. Perry				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546			10. SPONSORING / MONITORING AGENCY REPORT NUMBER NASA TM - 108409	
11. SUPPLEMENTARY NOTES Prepared by Structures and Dynamics Laboratory, Science and Engineering Directorate.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Unclassified—Unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Buildup of atmospheric trace contaminants in enclosed volumes such as a spacecraft may lead to potentially serious health problems for the crewmembers. For this reason, active control methods must be implemented to minimize the concentration of atmospheric contaminants to levels that are considered safe for prolonged, continuous exposure. Designing hardware to accomplish this has traditionally required extensive testing to characterize and select appropriate control technologies. Data collected since the Apollo project can now be used in a computerized performance simulation to predict the performance and life of contamination control hardware to allow for initial technology screening, performance prediction, and operations and contingency studies to determine the most suitable hardware approach before specific design and testing activities begin. The program, written in FORTRAN 77, provides contaminant removal rate, total mass removed, and per pass efficiency for each control device for discrete time intervals. In addition, projected cabin concentration is provided. Input and output data are manipulated using commercial spreadsheet and data graphing software. These results can then be used in analyzing hardware design parameters such as sizing and flow rate, overall process performance and program economics. Test performance may also be predicted to aid test design.				
14. SUBJECT TERMS atmospheric contamination, trace contaminant, control, software, analysis, analytical tool, spacecraft, systems design, systems analysis, ECLSS, atmospheric revitalization, computer program			15. NUMBER OF PAGES 173	
			16. PRICE CODE NTIS	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited	

ACKNOWLEDGMENTS

A complex computer model requires the expertise of many individuals to produce a useful engineering design tool. The trace contaminant control simulation computer program has benefited from the scientific, engineering, and computer programming expertise of many individuals since 1965. In particular, the granular activated charcoal performance and capacity characterization conducted by Dr. A.J. Robell and his colleagues, C.R. Arnold, A. Wheeler, G.J. Kersels, and R.P. Merrill, at Lockheed Missiles and Space Company in Sunnyvale, CA, formed the basis for the contaminant removal device routines. The work by Thomas Olcott, also of Lockheed, which built on the techniques developed by Dr. Robell, provided the first stepping stone for developing today's version of the computer program. The first generalized version was developed by R.B. Jagow, R.J. Jaffe, and R.A. Lamparter of Lockheed for analyzing Spacelab atmospheric contamination control designs. This work was expanded into an even more general model by Lamparter and R.Q. Yee to become the predecessor of the current program design. The work of Lamparter is further acknowledged for developing a more user-friendly version of the program which was the version that I began working on in 1989. These accomplishments were made under the direction of C.D. Ray of NASA's Marshall Space Flight Center. Without the basis set by these individuals, the current version would not exist. Development of version 8.0 also involved Wes Coleman of Sverdrup Technology in Huntsville, AL, who helped make the plot data output capabilities possible and provided a very nice solution to a convergence instability problem. He also provided a fresh viewpoint which helped find several bugs which I had overlooked. I am extremely grateful to Bob DaLee of McDonnell Douglas Space Systems Company in Huntsville, AL, who painstakingly translated the computer program from the IBM/compatible personal computer version to a version which is compatible with Apple Macintosh computers.

TABLE OF CONTENTS

	Page
INTRODUCTION	1
COMPUTER SIMULATION BACKGROUND	2
SIMULATION PROGRAM OVERVIEW	4
Computer Hardware Requirements.....	4
Input/Output Data Manipulation Recommendations	4
Calculation Overview.....	5
Input Data	5
Contaminant Removal Devices	5
Calculation Convergence.....	5
Output Data	6
PROGRAM OPERATION.....	6
Input Data File Preparation.....	8
Contaminant Data Input File Preparation	11
Removal Device Definition Input File Preparation.....	14
Device Type Number 1: The Cabin	14
Device Type Number 2: Cabin Leakage	16
Device Type Number 3: Axial Flow Charcoal Bed	17
Device Type Number 4: Radial Flow Charcoal Bed	17
Device Type Number 5: LiOH Bed	18
Device Type Number 6: Ambient Temperature CO Oxidizer.....	18
Device Type Number 7: High-Temperature Catalytic Oxidizer	18
Device Type Number 8: Condensing Heat Exchanger	19
Device Type Number 9: The Dummy Device	19
Removal Device Arrangement Rules	20
Removal Device Regeneration	21
Time-Dependent Data Input File Preparation.....	21
General Time-Dependent Change Rule	23
Changing Contaminant Generation Rates	23
Changing Device Definition Matrix Information	24
IBM and Compatible Program Version Execution	24
Reading Input Data	25
Data Output Decisions.....	25
Concentration and Device Efficiency Plot Data Output	26
Toxic Hazard Index Output	26
Standard Formatted Output	26
Plot Data File Designation.....	28
Calculation Routine Execution.....	28

TABLE OF CONTENTS (Continued)

	Page
Output Data File Manipulation	29
Standard Formatted Output	29
Concentration Plot Data	29
Efficiency Plot Data	29
Toxic Hazard Index Data	32
Macintosh Program Version Execution	32
 DETAILED COMPUTER PROGRAM DESCRIPTION	 35
Program Editing, Compiling, and Linking	35
Main Program	35
Calculation Loop Subroutines	37
Subroutine CAFILL	37
Subroutine RAFILL	37
Subroutine PCSET	39
Subroutine PRAFIL	40
Subroutine CNRSUB	40
Subroutine MASBAL	40
Subroutine CALCM	43
Subroutine LDIGEN	43
Subroutine PCAVCF	43
Subroutine RINCDD	45
Subroutine REGEN	45
Subroutine MCALC	45
Subroutine PREDCT	49
Subroutine LODEFF	49
Subroutine CONVRG	50
Subroutine SLIOH	50
Contaminant Removal Device Calculation Subroutines	50
Subroutine ACHBD	50
Reaction With Specially Treated Charcoals	56
Ammonia Removal	56
Formaldehyde Removal	56
Subroutine RCHBD	56
Subroutine ALIOH	57
Subroutine COOXID	57
Subroutine CATBNR	57
Subroutine CONDHX	57
Data Input and Output Subroutines	60
Subroutine CRIN	60
Subroutine RRIN	60
Subroutine CROUT2	60
Subroutine RROUT2	61
Subroutine CROUT	61

TABLE OF CONTENTS (Continued)

	Page
Subroutine RROUT	61
Subroutine DATOUT	61
Subroutine GROUP	61
Subroutine PRFANS	61
Subroutine HEADGS	61
APPENDIX A – Trace Contaminant Control Simulation (TCCS) Computer Program FORTRAN Code Listing	63
APPENDIX B – Toxic Hazard Index Description	159
REFERENCES	162

LIST OF ILLUSTRATIONS

Figure	Title	Page
1.	Simplified computer program flow chart	7
2.	Sample contaminant input data table and input file.....	12
3.	Sample device definition data	15
4.	Sample device configurations and dummy device use	20
5.	Several common time-dependent changes and a sample input file	30
6.	Example concentration plot data output	32
7.	Example Macintosh file selection window	33
8.	TCCS computer program user input screen	34
9.	Window for naming plot data output files	34
10.	Main program block flow diagram	36
11.	TCCS computer program subroutine listing and description	38
12.	Subroutine PCSET block flow diagram	39
13.	Subroutine CNRSUB block flow diagram	41
14.	Subroutine MASBAL block flow diagram	42
15.	Subroutine CALCM block flow diagram	44
16.	Subroutine RINCDD block flow diagram	46
17.	Subroutine REGEN block flow diagram	47
18.	Subroutine MCALC block flow diagram	48
19.	Subroutine PREDCT block flow diagram	49
20.	Subroutine CONVRG block flow diagram	51
21.	Charcoal saturation and adsorption zone distribution	52
22.	Potential plot for type BD granular-activated charcoal	53

LIST OF ILLUSTRATIONS (Continued)

Figure	Title	Page
23.	Potential plot showing relative humidity effects	54
24.	Adsorption zone length as a function of the potential factor	55
25.	Noble metal CO catalyst performance	58
26.	Absorption of contaminants by humidity condensate	59
27.	Ammonia solubility in water at varying carbon dioxide concentration	60

LIST OF TABLES

Table	Title	Page
1.	Atmospheric TCCS program development summary	3
2.	Required contaminant physical properties and generation data	8
3.	Example device definition table	9
4.	Device definition data by device type for data table columns 8 through 16	10
5.	Device type number and description	16
6.	Several common time-dependent changes	22
7.	A sample time-dependent data input file	23
8.	Example concentration plot data output.....	26
9.	Example removal device efficiency plot data output	27
10.	Example toxic hazard index plot data output	32

TECHNICAL MEMORANDUM

COMPUTERIZED ATMOSPHERIC TRACE CONTAMINANT CONTROL SIMULATION FOR MANNED SPACECRAFT

INTRODUCTION

Buildup of atmospheric trace contaminants in enclosed volumes such as a manned spacecraft may lead to potentially serious health problems for the crewmembers. For this reason, control methods are implemented to minimize the concentration of atmospheric contaminants to levels that are considered safe for prolonged, continuous exposure. Methods employed to achieve these levels are classified into two major categories—passive and active. Passive control refers to the material selection and control process to which all bulk materials and assembled articles must be subjected to qualify for flight onboard a manned spacecraft. Criteria included in this process are flammability, odor, and offgassing characteristics. Through this screening, the uncontrolled production via offgassing mechanisms of atmospheric contaminants from materials used in the habitable areas of the spacecraft are limited, but not eliminated. Active control methods must be used to provide the second level of contaminant control. Activities such as food preparation, housekeeping, and personal hygiene contribute to the base contamination load found in the spacecraft atmosphere. Contaminants are also produced by crewmember metabolism, experiment hardware and operations, and spacecraft hardware operations. The activities onboard the spacecraft combined with material offgassing produce contaminants in sufficient quantities that the atmospheric concentration may build up to levels which the crewmembers may find irritating or possibly intolerable. Therefore, the total contamination control approach requires both the passive and active approaches.

Active contamination control can be achieved through various means. Typical technologies used onboard manned spacecraft include physical adsorption on granular activated charcoal or other adsorbent material, chemical adsorption using impregnated granular activated charcoals and granular lithium hydroxide (LiOH), ambient temperature catalytic oxidation, and high temperature catalytic oxidation. Other means which are significant include removal by absorption into humidity condensate and spacecraft atmospheric leakage. These latter methods are not specifically designed to remove atmospheric contaminants, but have been shown to be significant when the total contaminant material balance in a spacecraft is considered.

Designs for active contamination control hardware have evolved from single granular-activated charcoal canisters used onboard earlier manned spacecraft, to a regenerable molecular sieve adsorbent used onboard the *Skylab* orbital workshop. Space shuttle and Spacelab programs use layered adsorbent canisters containing nonregenerable adsorbent and ambient temperature catalytic oxidation materials. Future contamination control hardware will build on these technologies to include regenerable adsorbent materials and high temperature oxidation catalysts.

Designing these contamination control devices is difficult since they must be designed without thorough knowledge of the overall load that must be controlled. Historical data from past manned missions are important for understanding the loads from offgassing, crew metabolism, and housekeeping sources. This information forms the basis for a generic spacecraft load model which can be used for hardware design purposes. As the spacecraft design matures, the contamination control hardware is refined by replacing the load model data with data for the actual flight hardware.

Ultimately, the spacecraft and contamination control designs converge. The final result is a robust contamination control hardware design with sufficient margin which allows a wide range of hardware and activities to be accommodated onboard the spacecraft with no adverse effects on the crewmembers' health.

COMPUTER SIMULATION BACKGROUND

In the past, designing contamination control hardware required extensive testing to characterize and select the appropriate control technologies. Capacities for granular-activated charcoal and other adsorbent materials had to be determined along with development of theoretical approaches to predicting the performance of future contamination control hardware. Early efforts to select technology and design hardware for spacecraft contamination control hardware began during the *Apollo* project. Studies conducted before 1965 by Lockheed Missiles and Space Company in Sunnyvale, CA, developed early spacecraft contaminant load models and selected candidate technologies.^{1 2 3} These technologies have been the basis for designing contamination control hardware for manned spacecraft ever since. More detailed studies on adsorption and catalytic oxidation followed. High temperature catalytic oxidation was studied by Olcott⁴ and contaminant adsorption by Robell, et al.⁵ by late 1970. In his work, Robell developed an extremely useful correlation for predicting the loading capacity of granular-activated charcoal by using the potential theory developed by Polanyi.⁶ This work became the basis for computer simulation of adsorption processes to be used onboard manned spacecraft for atmospheric contamination control. The first generation simulation program was developed by Olcott before May 1972 using the technique developed by Robell.⁷ By 1975, a prototype contamination control system had been fabricated by Lockheed, and the first generation computer simulation program for adsorption had been used under NASA contract NAS 1-11526 to analyze the hardware performance.⁸

Work continued in characterizing contamination control technologies, and low temperature oxidation catalysts were studied in 1977. That same year, the first integrated computer simulation model for spacecraft atmospheric contamination control was developed by Jagow, et al. at Lockheed.⁹ This program was specifically tailored for analyzing the performance of the Spacelab contamination control canister which consisted of two layers of adsorbent material followed by a layer of low-temperature, carbon monoxide oxidation catalyst. This program was modified and released later in 1977 as a general computer simulation program capable of simulating any spacecraft contamination control system given the appropriate input data.¹⁰ This basic computer program was used for design and preflight performance analyses for Spacelab missions.

Little work was conducted on the basic computer program until NASA contract NAS8-36406 produced a new release of the computer program in 1986 which featured input data manipulation using commercially available spreadsheet software and operation on a personal computer. This program version, designated version 5.0, was used as the basis for Space Station *Freedom* contamination control hardware design.¹¹ Improvements since 1986 have included further enhancements for input data manipulation, a routine to calculate the toxic hazard index, improved simulation of humidity condensate absorption, and improved output data manipulation flexibility. Written in FORTRAN 77, the program can be run on IBM-compatible personal computers (version 8.0 Alpha), Digital Equipment Corporation VAX mainframe computers using the VMS operation system (version 8.0 Beta), and Apple Computer, Inc., Macintosh® computers (version 8.0 Gamma). Table 1 summarizes the progression of the computer program from its initial version in 1970 to the present version which was completed in April 1992.

Table 1. Atmospheric TCCS program development summary.

Version	Release Date	Remarks
1.0	May 1972	Initial program version developed by Lockheed for assessing charcoal bed performance using empirical charcoal loading data. Documented in NASA contract report NAS CR-2027.
2.0	November 28, 1975	Improved program version used by Lockheed for prototype contamination control hardware performance predictions.
3.0	January 15, 1977	A generalized program developed by Lockheed based on versions 1.0 and 2.0 to predict Spacelab contamination control system performance.
4.0	September 1, 1977	A revised version of the Spacelab computer program which simulated more general contamination control systems designs. This version was developed by Lockheed.
5.0	September 5, 1986	The first version developed by Lockheed specifically for personal computer use. This version greatly enhanced data input capabilities.
6.0	June 1, 1989	Version 6.0 added the calculation of the toxic hazard index. This version was developed by the Boeing Company and MSFC personnel.
7.0 to 7.9	August 2, 1980 through April 1, 1992	Versions 7.0 through 7.9 were developed at MSFC. This progression improved the ability of the program to simulate transient contamination cases and an improved humidity condensate absorption routine based on Henry's law. Charcoal bed breakthrough calculation instability was corrected using a Newton-Raphson method and user interaction was improved.
8.0	April 10, 1992	Output routines were modified to allow data output for concentration, efficiency, and toxic hazard index to separate files which can be accessed by spreadsheet programs for data analysis and plotting. This version was developed at MSFC.

SIMULATION PROGRAM OVERVIEW

Computer Hardware Requirements

The trace contamination control simulation (TCCS) computer program is written in FORTRAN 77 and can be run on IBM AT-compatible personal computers running the disk operating system (DOS) version 3.0 or higher. This version requires at least an IBM AT or compatible machine with an 80287 floating point processor and a minimum of 512 kilobytes of memory. Upgraded personal computers using a 386 or 486 processor are preferred since they provide more rapid execution. The program is compiled and linked on the personal computer using Ryan-McFarland FORTRAN version 2.42.

A similar version of the program, version 8.0 Beta, allows operating on a Digital Equipment Corporation VAX mainframe computer which uses the VMS operating system. A 50,000 block disk allocation is recommended to accommodate files produced by a large simulation. The program files occupy up to 5,000 blocks of disk space. The mainframe version may offer increased speed and disk storage for very large simulations; however, recent improvements in personal computing speed and disk storage capacity have made this version less desirable especially since input data files are more easily created using a personal computer and uploaded to the mainframe computer. This makes a personal computer with file transfer capabilities an important hardware requirement for using the mainframe version.

Version 8.0 Gamma, which runs on Apple Computer, Inc., Macintosh machines, operates similarly to the Alpha and Beta versions. The Macintosh user interface, which uses a mouse and pop-up menus for input file management, makes the program even more user-friendly by requiring fewer commands to be entered from the keyboard. It is recommended that this version be run on a Macintosh II family machine with 8.0 megabytes of RAM and Macintosh System 7 operating system version 7.0 or later.

Input/Output Data Manipulation Recommendations

Input data is manipulated using commercially available spreadsheet software. The program can read input data using both space- and comma-delimited formats, while output data is produced in a space-delimited ASCII format. Because of these program input and output data formats, it is recommended that the Lotus Development Corporation (Lotus) software, Lotus 123® with Printgraph®, be used for input data generation and output data manipulation for the personal computer version. This software can print to a file in the space-delimited ASCII format required by the program and import program output data directly without any additional processing by the user. Other spreadsheet programs may be used, such as Microsoft Corporation's Excel®, for input file generation, but the user must save the input files in the comma separated variable (CSV) format. The ASCII-formatted output data produced by the program requires additional manipulation within the spreadsheet program, such as using the parse function, to allow it to be analyzed if Excel is used for output data analysis.

The mainframe program version input and output data manipulation is more difficult. It is recommended that input data files be prepared on a personal computer using Lotus 123® and uploaded to the mainframe. Text files may be created using the VMS editor; however, managing

these files is more difficult than using a spreadsheet program. Some output data, such as data to be used to generate plots, should be downloaded and manipulated using the personal computer spreadsheet software or other commercially available data plotting software.

The Macintosh version of the program should use the Microsoft Corporation Excel spreadsheet software for input file preparation and output data analysis. Input files prepared using Excel must be saved as a CSV file for the program to read the data. Output data produced by the program may be imported into the Excel spreadsheet and then the parse function run to place each number in a spreadsheet cell. Data plots may be prepared using Excel or the data may also be loaded into other popular Macintosh plotting software such as Cricket Graph® or Kaleidograph®.

Calculation Overview

The TCCS program calculates the cabin concentration of specified contaminants given a set of contamination control devices as a function of time. Depending on the output data selections made, data for each basic time increment may be output detailing the current contaminant cabin concentrations, their comparison to the spacecraft maximum allowable concentrations (SMAC's), the removal rate for each contaminant in each device, the cumulative total contaminant mass removed by each device, the increment device single pass removal efficiency for each contaminant, and the toxic hazard index.

Input Data

Input information consists of the names and properties of each contaminant, the order and design characteristics of the removal devices, and time dependent data such as changes in contaminant generation rate or removal device flow rate as a function of time. These input data are read into the program using three separate data files. The first file contains contaminant data, the second contains removal device definition data, and the third contains time dependent data. Files are used for input to minimize repetitive input by the user.

Contaminant Removal Devices

Removal device configuration is provided by a routine which allows up to 15 total devices with parallel and series arrangement. Up to three devices in parallel can be placed upstream of any removal device. The program simulates nine removal device types. Two device types, the cabin and leakage, must always be included in the device definition file. Modular programming allows for quick revision or addition of device types if necessary; however, any internal modifications require recompiling and relinking the program.

Calculation Convergence

A discrete time increment approach is used by the program which uses contaminant concentration from the previous increment as the initial concentration. The final concentration is calculated based on the device configuration, contaminant generation rate, contaminant removal rate, removal device efficiency, and other specific information relevant to that time increment. This approach allows flexibility which simulates transient situations such as inadvertent contaminant releases or other time dependent contamination events. The calculation data are stored in matrices which increases the program execution speed.

Since contaminant properties can vary with respect to ease of removal and cabin concentration, small time increments would be the most accurate for any given simulation. However, the program uses a log-mean direct solution approach combined with a Newton-Raphson convergence routine to allow much larger time increments which significantly reduce the computer runtime required. This approach produces an exact solution during steady-state conditions with constant removal device efficiencies. This is normally the case during a simulation, but efficiency may vary in some cases, such as for a contaminant that is breaking through an adsorbent bed. In this case, the Newton-Raphson convergence technique combined with decreasing the time increment to one-twentieth the size of the basic increment for that contaminant only provides the desired calculation accuracy without affecting the overall run time.

The prediction and convergence routines use the Newton-Raphson technique to ensure that the removal device calculations are made with the proper contaminant inlet concentration. Contaminant concentrations are predicted using removal efficiencies of the previous increment and the generation rates for the present increment. Actual device efficiencies and contaminant concentrations are calculated by the convergence routine and compared to the predicted values. This process continues until the predicted and calculated values agree within a specified error.

If convergence is not obtained for a contaminant, warning statements built into the program alert the user. In addition, other warning statements concerning input and output are provided to aid simulation diagnostics. Print options are provided to allow printing the contents of calculation matrices and convergence values. These options allow quick identification of problems with the input device configuration or contaminant data.

Output Data

Program output data consist of standard formatted output with column headings and page numbers and interactive formatted output to files for data plotting. Standard formatted data output is specified using print switches in the device definition input file while plot data output is specified interactively. Standard output consists of the contaminant name, cabin concentration, SMAC, and a designation of whether the cabin concentration exceeds the SMAC. Also, the standard print switches allow the contaminant removal rate, cumulative mass removed, and removal efficiencies for each contaminant in each device to be output. Standard output may be directed to the computer terminal screen, a specific data file, or a printer. These data include page numbers, time increment identification, and other information relating to the time increment. Interactive output decisions are made by the user to produce separate files containing concentration, device efficiency, and toxic hazard index data as a function of time. These files are in a spreadsheet format and can be imported to a spreadsheet to allow manipulation and preparation of data plots. The toxic hazard index is also interactively sent to the standard formatted output in addition to a plot file if desired.

PROGRAM OPERATION

The computerized atmospheric trace contaminant control simulation program requires 33 FORTRAN 77 routines. Figure 1 shows a block flow diagram of a simulation of an atmospheric contamination control case. An operation description for using this program includes instructions for preparing input data files and operating the program.

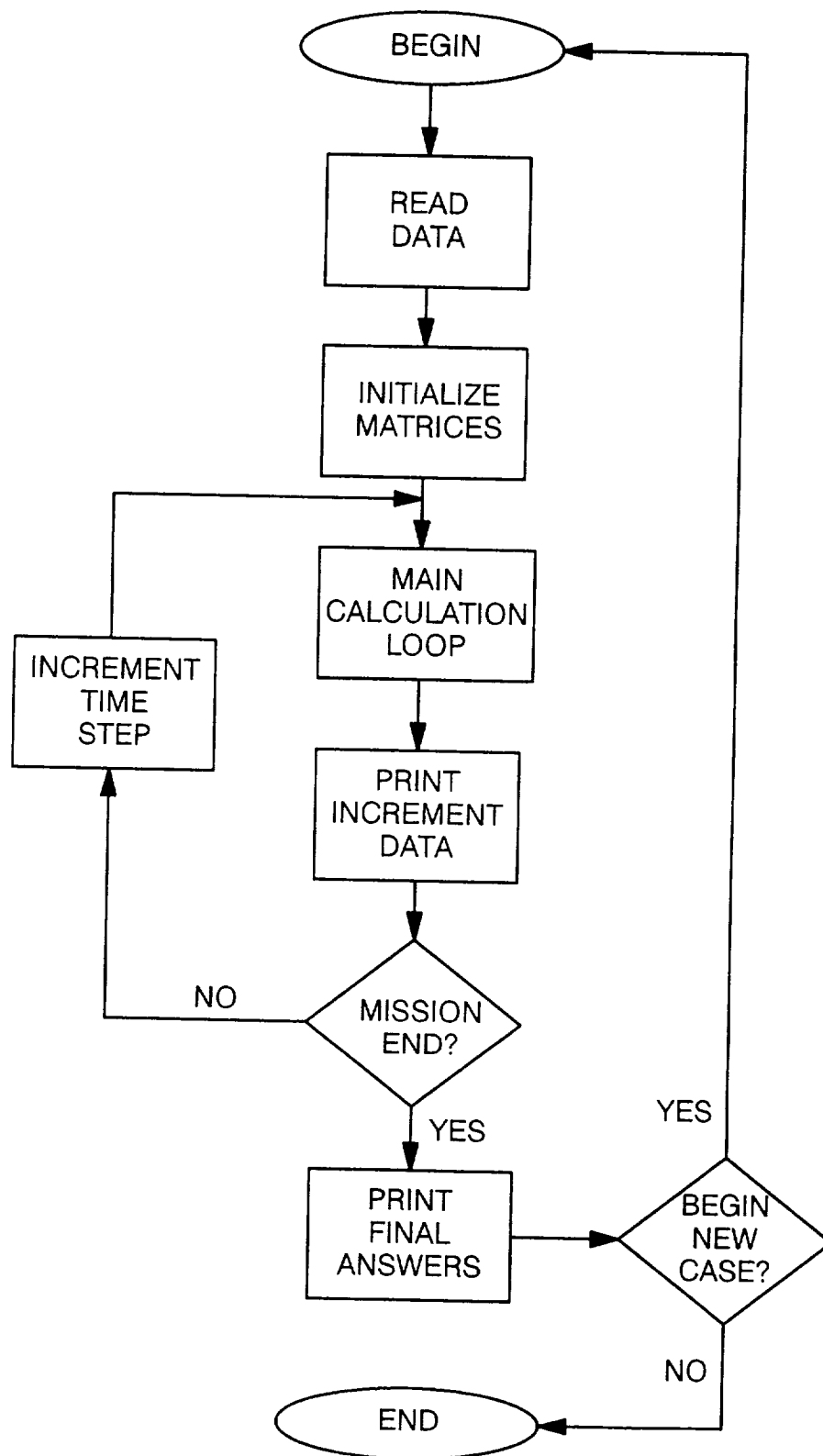


Figure 1. Simplified computer program flow chart.

Input Data File Preparation

Before program operation, three input files must be prepared to define the contaminant properties and generation rates, removal device configuration, and time dependent inputs. These files are created using commercially available spreadsheet software which is capable of producing space delimited ASCII files. Lotus 123 is the preferred spreadsheet program for creating these input files. The print-to-file option in Lotus is used to send the appropriate data range to each file. These files are read by the program, and the data is placed into calculation matrices NN, CDI, DD, and TT. Matrix NN is a 1 by 300 matrix containing the contaminant names, CDI is a 23 by 300 matrix containing contaminant physical properties defined in table 2, matrix DD is a 15 by 22 matrix containing removal device design and configuration data defined in tables 3 and 4, and matrix TT is a 7 by 300 matrix containing time dependent contaminant generation rate and device flow rate data. The matrix sizes are adjustable; however, the personal computer version is limited to 300 contaminants so that the contaminant matrix does not exceed 64 kilobytes to provide a more rapid run time.

Table 2. Required contaminant physical properties and generation data.

Contaminant Property	Units and Remarks
Contaminant name	30 character maximum
Cabin generation rate	mg/h
Liquid density	gm/cm ³ (cabin conditions)
Molar volume	cm ³ /gram-mole (cabin conditions) (0 = no charcoal removal)
Molecular weight	gm/gram-mole
Vapor pressure	mg/m ³ (cabin conditions)
Henry's Law constant	atm/mole-fraction
Lithium hydroxide removal	lb _m LiOH/lb _m contaminant removed (0 = no removal)
Chemical category	According to NHB 8060.1B Appendix D
Maximum allowable concentration	mg/m ³
Generation rate in devices 3 to 15	mg/h
Degree of oxidation in a high-temperature oxidizer	0 = none; 1 = fully

Table 3. Example device definition table.

TCCS PROGRAM INPUT DATA															
DEVICE DEFINITION TABLE															
=====															
MATRIX COLUMN NO.															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

DEVICE NO.	DEVICE FLOW (m3/h)	DEVICE TYPE NO.	UPSTREAM DEVICES BY NUMBER NO.1	NO.2	NO.3	NOT FOR INPUT	COL 8 TO 16 USED FOR DEVICE SPECIFIC INPUTS SEE TABLE BELOW								

1	0.00	1	0	0	0	0	0	227.86	293	24	1.00E-01	1.2	0	0	0
2	0.0460	2	1	0	0	0	1	0	0	0	0	0	1	1	0
3	519.90	8	1	0	0	0	1	1.451	0	0	0	0	0	0	0
4	15.12	3	1	0	0	0	1	0.580	0.320	0	490	2	0	0	0
5	4.08	7	4	0	0	0	1	0.000	0.000	0	0	0	0	0	0
6	4.08	5	5	0	0	0	1	0.234	0.129	0	442	0	0	0	0

Table 4. Device definition data by device type for data table columns 8 through 16.

Device Number		Device Type	Device Definition Table Column Number												
			8	9	10	11	12	13	14	15	16				
1		Cabin	0	Volume (m ³)	Temperature (K)	Time Increment (h)	Converge Error (decimal)	Coexist Factor (1.20)	0						
2		Leakage	1	Print Switch No.1	Print Switch No.2	Print Switch No. 3	Print Switch No. 4	Print Switch No. 5	Print Switch No. 6	Print Switch No. 7	Print (3) Switch No. 8				
3		Axial Charcoal	Maximum Efficiency (decimal)	Bed Length (m)	Bed Diameter (m)	0	Matl. Bulk Density (kg/m ³)	Bed Treatment Type (4)	Regeneration Data						
4		Radial Charcoal	Maximum Efficiency (decimal)	Bed Length (m)	Bed Outside Diameter (m)	Bed Inside Diameter (m)	Matl. Bulk Density (kg/m ³)	Bed Treatment Type (4)	Regeneration Data						
									First (h)	Interval (h)	Duration (h)				
5		LiOH Bed	Maximum Efficiency (decimal)	Bed Length (m)	Bed Diameter (m)	0	Matl. Bulk Density (kg/m ³)	Changeout Data		0		0		0	
6		CO Oxidizer	Maximum Efficiency (decimal)	Bed Length (m)	Bed Diameter (m)	0	0	0	0	0	0	First (h)	Interval (h)	Duration (2)	
												Interval (h)			
7		Catalytic Oxidizer	1	0	0	0	0	0	0	0	0	0	0	0	
8		Condensing Heat Exchanger	1	Condensate Flow (kg/h)	0	0	0	0	0	0	0	0	0	0	
9		Dummy Device	0	0	0	0	0	0	0	0	0	0	0	0	

- NOTES:
1. Used to store total weight of LiOH used in the calculation in kg/h
 2. Used to store total weight of LiOH used during the calculation in kg
 3. Print switches are set to 1 for printing and 0 for no printing. No. 1 to 5 for diagnostics, 6 to 8 for printout.
 4. 0 = no treatment; 1 = chromate treatment for formaldehyde; 2 = phosphoric acid treatment for ammonia

Contaminant Data Input File Preparation

The contaminant data input file contains the contaminant names, physical properties, generation rates in the cabin and each device, and the degree of oxidation in a high temperature catalytic oxidation process. These data are read into program calculation matrices NN and CDI. Sample input is shown in figure 2. Data required for each column of the contaminant data input file are the following:

1. The contaminant sequential number, 1 through 300
2. An apostrophe to enclose the contaminant name as required by FORTRAN 77
3. The contaminant name up to 30 characters
4. An apostrophe to enclose the contaminant name
5. The contaminant cabin generation rate in milligrams per hour (mg/h)
6. The contaminant liquid phase density in grams per cubic centimeter (g/cm^3)
7. The contaminant molar volume in cubic centimeters per gram-mole (cm^3/gmole)
8. The contaminant molecular weight in grams per gram-mole (g/gmole)
9. The contaminant vapor pressure at the cabin temperature expressed in milligrams per cubic meter (mg/m^3)
10. The water solubility for the contaminant expressed as the Henry's Law coefficient in atmospheres per mole fraction (atm/mole fraction)—enter a zero if the compound is insoluble or no Henry's Law coefficient is available
11. The LiOH removal expressed as the weight of LiOH consumed per weight of contaminant removed in $\text{lb}_m \text{LiOH}/\text{lb}_m \text{contaminant}$
12. The chemical group classification for the contaminant specified in appendix D of NHB 8060.1B
13. The SMAC or other threshold value expressed in milligrams per cubic meter (mg/m^3)
14. The contaminant generation rate at the inlet to device number 3 in mg/h
15. The contaminant generation rate at the inlet to device number 4 in mg/h
16. The contaminant generation rate at the inlet to device number 5 in mg/h
17. The contaminant generation rate at the inlet to device number 6 in mg/h
18. The contaminant generation rate at the inlet to device number 7 in mg/h

CONTAMINANT INPUT DATA TABLE (Spacelab Missions 1 and 3 basis using maximum generation rates)

MAKE SURE ALL CELLS HAVE A VALUE OR 0 IN CELL

CONT		NAME(30 CHR MAX)		CABIN LIQUID		MOLAR		MOLAR		VAPOR		H2O SOL		LIOH		NHB		SMAC		GENERATION IN DEVICES 3 THROUGH 15															OXIDIZER	
NO.		RATE	DENSITY	VOL	g/cc	cc/mol	g/mol	MASS	MOLAR	CONC	mg/m3	atm	Henry	lb	lb	cont	GROUP	mg/m3		mg/h															decimal	EFF
1	1,1,2-triCl-1,2,2-trifluethane	279.99	1.564	120.0	187.40	3280000	2.69E+04	0	7	383.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.33			
2	1,3-Butadiene	0.06	0.676	82.0	54.09	7047000	2.42E+03	0	9	221.20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00			
3	Methane	30.81	0.425	37.7	16.04	225400000	3.54E+04	0	9	1771.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.95			
4	2-butanone	45.81	0.805	96.7	72.11	405400	1.63E-01	0	11	59.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00			
5	1-benzo[b]pyrrole	1.04	1.220	139.8	117.15	74	2.48E-02	0	15	0.48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00			
6	methyl hydrazine	0.03	0.875	63.0	46.07	1233000	2.70E-02	0	15	0.08	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00			
7	Ammonia	53.49	0.639	25.0	17.00	7522000	9.37E-01	0	16	17.40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00			
8	Carbon monoxide	24.04	0.803	22.1	28.01	974700000	6.34E+04	0	16	28.60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00			
9	Hydrogen	4.65	0.071	3.0	2.02	2033000	6.52E+04	0	16	247.30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00			

CONTAMINANT INPUT DATA FILE EXAMPLE

```

1 1,1,2-triCl-1,2,2-trifluethane 279.99 1.564 120.0 187.40 3280000 2.69E+04 0 7 383.00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.33
2 1,3-Butadiene 0.06 0.676 82.0 54.09 7047000 2.42E+03 0 9 221.20 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1.00
3 Methane 30.81 0.425 37.7 16.04 225400000 3.54E+04 0 9 1771.00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.95
4 2-butanone 45.81 0.805 96.7 72.11 405400 1.63E-01 0 11 59.00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1.00
5 1-benzo[b]pyrrole 1.04 1.220 139.8 117.15 74 2.48E-02 0 15 0.48 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1.00
6 methyl hydrazine 0.03 0.875 63.0 46.07 1233000 2.70E-02 0 15 0.08 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1.00
7 Ammonia 53.49 0.639 25.0 17.00 7522000 9.37E-01 0 16 17.40 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1.00
8 Carbon monoxide 24.04 0.803 22.1 28.01 974700000 6.34E+04 0 16 28.60 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1.00
9 Hydrogen 4.65 0.071 3.0 2.02 2033000 6.52E+04 0 16 247.30 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1.00

```

NOTE: THE CONTAMINANT INPUT DATA FILE IS CREATED BY WRITING THE SELECTED RANGE TO A FILE FROM THE SPREADSHEET. THIS FILE IS IN ASCII FORMAT.
THE CONTAMINANT NUMBER IS NOT SELECTED FOR THE INPUT DATA FILE.

Figure 2. Sample contaminant input data table and input file.

19. The contaminant generation rate at the inlet to device number 8 in mg/h
20. The contaminant generation rate at the inlet to device number 9 in mg/h
21. The contaminant generation rate at the inlet to device number 10 in mg/h
22. The contaminant generation rate at the inlet to device number 11 in mg/h
23. The contaminant generation rate at the inlet to device number 12 in mg/h
24. The contaminant generation rate at the inlet to device number 13 in mg/h
25. The contaminant generation rate at the inlet to device number 14 in mg/h
26. The contaminant generation rate at the inlet to device number 15 in mg/h
27. The decimal oxidation efficiency per pass for the contaminant in a high temperature catalytic oxidizer where 1.0 is fully oxidized and 0.0 is no oxidation.

Some general considerations should be noted when preparing the contaminant data input file. They are the following:

1. The molecular weights for ammonia, formaldehyde, carbon monoxide, and hydrogen must be exactly 17.0, 30.03, 28.10, and 2.02 since they are used as flags in the program
2. The entire set of information, columns 1 through 27, must be input for each individual contaminant
3. Each cell in all matrices (contaminant, device, and time) must contain a value or a zero
4. The column width in all matrices (contaminant, device, and time) should be sufficient to permit at least one blank space between the column entries
5. Printing to a file should be unformatted and the range should exclude the first column (contaminant number) and any column headings
6. The file must be created in one piece without any line feeds (except at the end of a line) or carriage returns and the left margin must be set to 0 and the right margin to 240 when using Lotus 123
7. The contaminant generation rates at the inlets of devices (columns 14 through 26) are for cases where contaminants are generated in a closed or confined space and are passed through a removal device(s) before entering the cabin (for example, a contaminant is produced in a experimental working volume and is passed through a charcoal bed before entering the cabin)
8. Some contaminant physical property data may not be available in the literature and must be estimated using correlations.

Removal Device Definition Input File Preparation

The device definition matrix, DD, contains the information which defines the removal device configuration. This information includes the type of devices, their arrangement, flow rates, and physical size or dimensions. Basic system data; such as the cabin volume, temperature, basic time increment size, convergence error specification, and the multicomponent adsorption coexistence factor; are also contained in this matrix. In addition, program data output for standard formatted data is controlled from this matrix. A sample device definition file is shown by figure 3. Data required for each of the first seven columns are the following:

1. The device number ranging from 1 through 15
2. The standard volumetric atmospheric flow through the device in cubic meters per hour (m^3/h)
3. Device type number as defined in table 5
4. Upstream device number 1 by device number (column 1)
5. Upstream device number 2 by device number
6. Upstream device number 3 by device number
7. A zero is entered in this column for all devices since this column is used to store calculated values during the program run.

Columns 8 through 16 are specific for each device type. Data required for these columns for each device type are the following:

Device Type Number 1: The Cabin

8. Device decimal removal efficiency equal to zero
9. Cabin atmospheric volume in cubic meters (m^3)
10. Cabin absolute temperature in kelvin (K)
11. Basic time increment in hours (h)
12. Convergence error as decimal percent (recommended values of 0.01 or 0.001)
13. Coexistence factor, approximate value of 1.20, to account for interaction or blockage of contaminants in granular activated charcoal beds
14. Not used for input therefore zero is input
15. Not used for input therefore zero is input
16. Not used for input therefore zero is input.

Example Device Definition Data Table

TCCS PROGRAM INPUT DATA																
USE RANGE 'LIST' DATA TABLE TO PRINT TO PRINTER										DEVICE DEFINITION TABLE						
USE RANGE 'DATA' TO PRINT DATA TO FILE																
=====																
MATRIX COLUMN NO.																
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
DEVICE NO.		DEVICE FLOW (m3/h)	DEVICE TYPE	UPSTREAM DEVICES BY NUMBER			NOT FOR INPUT		COL 8 TO 16 USED FOR DEVICE SPECIFIC INPUTS							
			NO.	NO. 1	NO. 2	NO. 3	SEE TABLE BELOW									

1	0.00	1	0	0	0	0	0	227.86	293	24	1.00E-01	1.2	0	0	0	0
2	0.0460	2	1	0	0	0	1	0	0	0	0	0	1	1	0	0
3	519.90	8	1	0	0	0	1	1.451	0	0	0	0	0	0	0	0
4	15.12	3	1	0	0	0	1	0.580	0.320	0	490	2	0	0	0	0
5	4.08	7	4	0	0	0	1	0.000	0.000	0	0	0	0	0	0	0
6	4.08	5	5	0	0	0	1	0.234	0.129	0	442	0	0	0	0	0

Example Device Definition Input File

1	0.00	1	0	0	0	0	0	227.86	293	24	1.00E-01	1.2	0	0	0	0
2	0.0460	2	1	0	0	0	1	0	0	0	0	0	1	1	0	0
3	519.90	8	1	0	0	0	1	1.451	0	0	0	0	0	0	0	0
4	15.12	3	1	0	0	0	1	0.580	0.320	0	490	2	0	0	0	0
5	4.08	7	4	0	0	0	1	0.000	0.000	0	0	0	0	0	0	0
6	4.08	5	5	0	0	0	1	0.234	0.129	0	442	0	0	0	0	0

Figure 3. Sample device definition data.

Table 5. Device type number and description

Type Number	Description
1	Cabin
2	Cabin leakage
3	Axial flow charcoal bed
4	Radial flow charcoal bed
5	Lithium hydroxide bed
6	Carbon monoxide oxidizer
7	High temperature catalytic oxidizer
8	Condensing heat exchanger
9	Dummy device

Device Type Number 2: Cabin Leakage

8. Device decimal removal efficiency equal to 1.0.

Columns 9 through 16 for the leakage device are used to set standard formatted data output print switches. These columns contain either zero (0) to tell the program not to print or one (1) to tell the program to print. Print switches 1 through 5 are for diagnostic data output and print switches 6 through 8 are for final data output. Typically, print switches 1 through 5 are set to zero during a normal simulation run.

9. Set print switch 1 to zero or one to regulate printing the contents of matrices CC and DD at the end of the precalculation setup routine, PCSET
10. Set print switch 2 to zero or one to regulate printing of the one-tenth increment convergence routine for one time increment for one contaminant (these results will be printed only if the one-tenth increment routine is used)
11. Set print switch 3 to zero or one to regulate printing of the convergence values of predicted and calculated contaminant concentration for the convergence routine CONVRG for the standard and one-twentieth time increments (useful if convergence problems occur)
12. Set print switch 4 to zero or one to regulate printing of the results for one contaminant from the main calculation routine, MCALC, after the average concentration is calculated (matrix DD and part of matrix CC for one contaminant)
13. Set print switch 5 to zero or one to regulate printing the contents of matrices DD and CC at the end of each time increment
14. Set print switch 6 to zero to print only the cabin concentration for each contaminant for each time increment or to one to print the contaminant removal rate, mass removed, and device efficiencies in addition to the cabin concentration for each time increment

15. Set print switch 7 to zero to allow form feeds between output pages or to one to suppress form feeds between output pages
16. Set print switch 8 to zero to print calculation results specified by print switch 6 at the end of the run only or to one to print calculation results at the end of each time increment.

Device Type Number 3: Axial Flow Charcoal Bed

8. Maximum device decimal removal efficiency equal to 1.0
9. Charcoal bed length in meters (m)
10. Charcoal bed diameter in meters (m)
11. Not used for input and set to 0
12. Charcoal bulk density in kilograms per cubic meter (kg/m^3)
13. Charcoal bed treatment type (1 for chromate treated charcoal for formaldehyde removal, 2 for phosphoric acid treated charcoal for ammonia removal, or 0 for untreated charcoal)
14. Time of first regeneration in hours (h)
15. Time interval between each regeneration in hours (h)
16. Regeneration duration in hours (h).

Device Type Number 4: Radial Flow Charcoal Bed

8. Maximum device decimal removal efficiency equal to 1.0
9. Radial flow bed length in meters (m)
10. Radial flow bed outside diameter in meters (m)
11. Radial flow bed inside diameter in meters (m)
12. Charcoal bulk density in kilograms per cubic meter (kg/m^3)
13. Charcoal bed treatment type (1 for chromate treated charcoal for formaldehyde removal, 2 for phosphoric acid treated charcoal for ammonia removal, or 0 for untreated charcoal)
14. Time of first regeneration in hours (h)
15. Time interval between each regeneration in hours (h)
16. Regeneration duration in hours (h).

Device Type Number 5: LiOH Bed

8. Maximum device decimal removal efficiency equal to 1.0
9. LiOH bed length in meters (m)
10. LiOH bed diameter in meters (m)
11. Not used for input and set to 0
12. LiOH bulk density in kilograms per cubic meter (kg/m^3)
13. Time of first changeout in hours (h)
14. Time between changeouts in hours (h)
15. Not used for input and set to 0
16. Not used for input and set to 0.

Device Type Number 6: Ambient-Temperature CO Oxidizer

8. Maximum device decimal removal efficiency equal to 1.0
9. Oxidizer bed length in meters (m)
10. Oxidizer bed diameter in meters (m)
11. Not used for input and set to 0
12. Not used for input and set to 0
13. Not used for input and set to 0
14. Not used for input and set to 0
15. Not used for input and set to 0
16. Not used for input and set to 0.

Device Type Number 7: High-Temperature Catalytic Oxidizer

8. Maximum device decimal removal efficiency equal to 1.0
9. Not used for input and set to 0
10. Not used for input and set to 0
11. Not used for input and set to 0

- 12. Not used for input and set to 0
- 13. Not used for input and set to 0
- 14. Not used for input and set to 0
- 15. Not used for input and set to 0
- 16. Not used for input and set to 0.

Device Type Number 8: Condensing Heat Exchanger

- 8. Maximum device decimal removal efficiency equal to 1.0
- 9. Condensate flow rate in kilograms per hour (kg/h)
- 10. Not used for input and set to 0
- 11. Not used for input and set to 0
- 12. Not used for input and set to 0
- 13. Not used for input and set to 0
- 14. Not used for input and set to 0
- 15. Not used for input and set to 0
- 16. Not used for input and set to 0.

Device Type Number 9: The Dummy Device

The dummy device serves to set the flow rate through a line when combining or splitting flows between upstream and downstream devices. Legal and illegal dummy device use is illustrated at the bottom of figure 4. When combining or splitting streams, dummy devices are required to set the upstream flow rates being combined or split.

- 8. Maximum device decimal removal efficiency equal to 0
- 9. Not used for input and set to 0
- 10. Not used for input and set to 0
- 11. Not used for input and set to 0
- 12. Not used for input and set to 0
- 13. Not used for input and set to 0

14. Not used for input and set to 0
15. Not used for input and set to 0
16. Not used for input and set to 0

Removal Device Arrangement Rules

Some simple rules apply to arranging removal devices in the removal device definition file. These rules are the following:

1. The first device must always be the cabin device type 1
2. The second device must always be the leakage device type 2
3. Device numbers 3 through 15 may be any legal device type
4. Devices must be numbered so that calculations for upstream devices occur before calculations for downstream devices since calculations are sequential by the device number specified in column 1
5. Inlet flow rates must be specified for each device to allow proper inlet concentration calculation for split flows
6. If the cabin is upstream of any device, column 4 should indicate this by entering a 1.

Figure 4 shows a sample device configuration along with some legal and illegal device configurations.

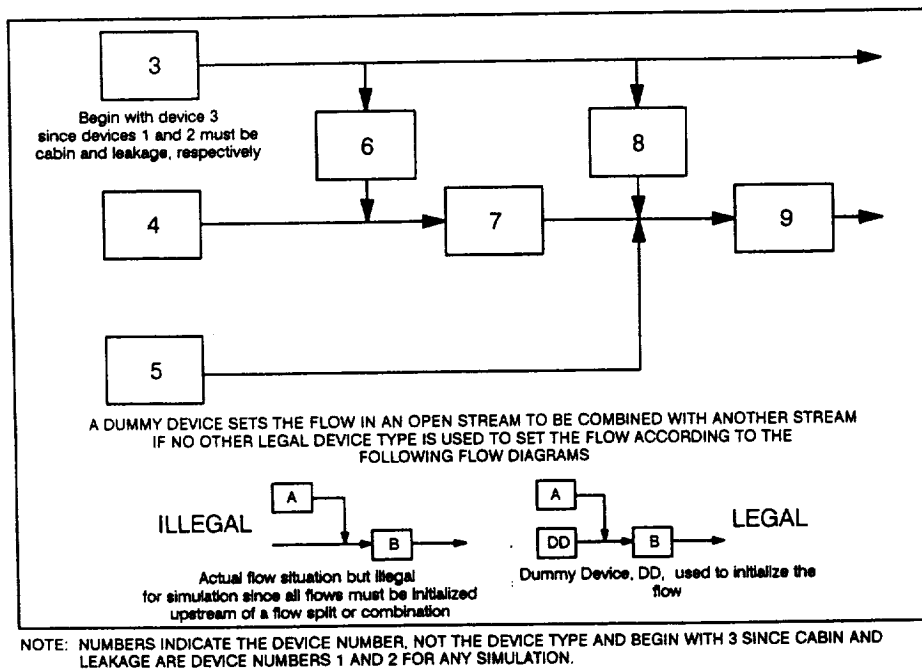


Figure 4. Sample device configurations and dummy device use.

Removal Device Regeneration

Regeneration of the axial and radial flow charcoal beds and the LiOH bed are accomplished by entering the appropriate regeneration information into matrix DD. Columns 14, 15, and 16 are used for regenerating the axial and radial charcoal beds and should contain the first time in hours from the beginning of the run that regeneration occurs (note that the run begins at zero hours and the run time increments itself according to the basic time increment), the time in hours between each regeneration, and the time required to regenerate the bed in hours. Columns 13 and 14 are used for LiOH bed regeneration and should contain the first changeout time in hours and the time between each change out in hours, respectively. No regeneration duration is required for the LiOH bed. It is important that the time between each regeneration or changeout and the regeneration time must be equal to or an even multiple of the basic time increment for the calculation. If they are not, the program will truncate them. Rules to be followed when simulating a regenerable bed are the following:

1. The initial regeneration time, changeout or regeneration interval, and time to regenerate must be multiples of the basic time increment specified in the device definition matrix row 1 column 11
2. If the values are not even multiples of the time increment, they will be truncated to the next smallest multiple which could be zero
3. If the regeneration duration desired is less than one time increment, the input flow rate can be modified manually in the program input according to the following:

$$Q_{\text{mod}} = Q_{\text{actual}} \times \{1 - (\tau_{\text{reg duration}} / \tau_{\text{reg interval}})\} \quad (1)$$

The modified flow rate should be input in the device definition matrix column 2 for that device. the regeneration duration should then be set to zero. If the flow rate is changed during the simulation, it may be necessary to add a dummy device in parallel with the regenerating device so that the original flow rate may be supplied to any possible downstream devices

4. For no regeneration, set the regeneration interval equal to zero.

Time-Dependent Data Input File Preparation

During a simulation run, it is possible to change the generation rate of any contaminant both in the cabin or for any device which is operating. Device flow rates and other device information such as the cabin volume, the cabin temperature, and basic time increment may also be changed. Time-dependent changes are accomplished by using the time dependent data matrix, TT. This matrix contains seven columns for each row. Each line in matrix TT does not allow simultaneous contaminant generation rate and device definition data changes. For example, the flow rate and contamination generation rate cannot be changed using a single line. Each change must have its own unique change line. However, two simultaneous device definition changes may be made for a single removal device using one time change file line. Table 6 shows the heading for the time-dependent data table and six common cases for time-dependent data changes. An example time-dependent data file is shown by the seven columns under the data heading of table 7. The time-dependent data file read by the program is created by selecting only the time change numbers when writing the data to the file. The

Table 6. Several common time-dependent changes.

Type of Change	Change Items						
Time-Dependent Data Table Sample Heading	FILE:TCTT02 28-May 1992 TIME DEPENDENT CONTAMINANT INPUT DATA FILE (MAT TT) MAKE SURE ALL CELLS HAVE VALUE OR 0 IN CELL USE RANGE 'LIST' TO PRINT TO PRINTER USE RANGE 'DATA' TO PRINT TO FILE =====						
	CHANGE CONT GEN IN MAT CDI <-----> CHANGE DEVICE FLOW RATE IN MAT DD <-----> CHANGE OTHER MAT DD INFO <----->						
	CHANGE TIME (h)	CONT NO.	CONT GEN RT (mg/h)	DEVICE NO.	NEW Q DEVICE (m ³ /h)	MAT DD COLUMN NO.	NEW VALUE
1. No changes	0	0	0	0	0	0	0
2. Change contaminant generation rate	24	1	800	1	-1	-1	-1
3. Change device flow rate	24	-1	-1	3	15	-1	-1
4. Change cabin volume	24	-1	-1	1	-1	9	150
5. Change device flow rate and bed length	24	-1	-1	3	15	9	1.5
6. Change time increment	24	-1	-1	1	-1	11	1

Table 7. A sample time-dependent data input file.

Change Type	Data to be Read by the Computer Program						
Change increment	720	-1	-1	1	-1	11	1
Change flow	720	-1	-1	3	4.8	-1	-1
Change flow	720	-1	-1	4	4.8	-1	-1
Change flow	720	-1	-1	5	4.8	-1	-1
Change increment	732	-1	-1	1	-1	11	6
Change increment	768	-1	-1	1	-1	11	5
Change increment	773	-1	-1	1	-1	11	23
Change flow	773	-1	-1	3	0	-1	-1
Change flow	773	-1	-1	4	0	-1	-1
Change flow	773	-1	-1	5	0	-1	-1
Change increment	796	-1	-1	1	-1	11	24
Change increment	844	-1	-1	1	-1	11	2
Change flow	846	-1	-1	3	4.8	-1	-1
Change flow	846	-1	-1	4	4.8	-1	-1
Change flow	846	-1	-1	5	4.8	-1	-1
Change flow	846	-1	-1	6	281	-1	-1
Change generation	846	5	0.1274	1	-1	-1	-1
Change generation	846	8	3.4825	1	-1	-1	-1
Change generation	846	13	3.5398	1	-1	-1	-1
Change generation	846	14	2.8109	1	-1	-1	-1
Change generation	846	22	0.3228	1	-1	-1	-1
Change generation	846	27	0.6651	1	-1	-1	-1
Change generation	846	121	14.9589	1	-1	-1	-1
Change generation	846	137	4.2861	1	-1	-1	-1
Change generation	846	149	0.0075	1	-1	-1	-1
Change generation	846	153	39.6206	1	-1	-1	-1
Change generation	846	154	6.5515	1	-1	-1	-1
Change generation	846	155	2.1837	1	-1	-1	-1
Change increment	892	-1	-1	1	-1	11	24

heading should not be included in the data file. If it is, the program will write an error message to the screen telling the user that the data file cannot be read. Each time-dependent file may contain up to 300 lines of changes. Rules for using this data file are the following:

General Time-Dependent Change Rule

1. Column 1 contains the change time in hours for all changes. Any update can occur only at the beginning of a time increment. If a change time is not an exact multiple of the basic time increment, the time will be truncated to the next lowest multiple of the increment.

Changing Contaminant Generation Rates

1. Column 1 contains the change time in hours (h)

2. Column 2 contains the contaminant sequential number to be changed. The contaminant number must be greater than or equal to one, and less than or equal to the maximum number of contaminants in the contaminant input file
3. Column 3 contains the new or changed generation rate
4. Column 4 contains the removal device number whose contaminant generation rate is to be changed. The device number must be greater than or equal to 1, and less than or equal to the maximum number of devices in the simulation. (Device 2 may be used, but it does not effect the calculation since any change would be canceled out by the 100-percent removal efficiency provided by cabin leakage.)
5. Column 5 is ignored and is set equal to -1
6. Column 6 is ignored and is set equal to -1
7. Column 7 is ignored and is set equal to -1.

Changing Device Definition Matrix Information

1. Column 1 contains the change time in hours (h)
2. Column 2 is ignored and is set equal to -1
3. Column 3 is ignored and is set equal to -1
4. Column 4 contains the device number and must be greater than or equal to 1 and less than or equal to the maximum number of removal devices
5. Column 5 contains the new device definition matrix information. This information must be greater than zero if a change is desired. If no input is desired, use -1 since this will prevent the program from reading the new information in column 7
6. Column 6 contains the device definition matrix column number that the change is to occur and must be greater than or equal to zero and less than or equal to 16. Any other value in this column causes the program to ignore the value in column 7
7. Column 7 contains the new device definition matrix value. Both the flow rate and any other device change can be made simultaneously for any device.

A time-dependent data file is necessary for program execution even when no time dependent changes will be made. In this case, a single row with all the columns set equal to zero is required for the time-dependent data file.

IBM and Compatible Program Version Execution

Once the three input files have been prepared, the TCCS computer program is ready to be executed. Execution is initiated by typing TCCS at the DOS prompt. If a very large contaminant data

file will be used for the simulation and contaminant plot data will be obtained during the run, the program should be executed by typing TCCS /R 10000 to allow for larger output records during the run. The number used with the "/R" option can range between 512 and 65,527 and corresponds to record lengths of 512 bytes to 65,527 bytes. Normal execution without the "/R" option results in the default record length of 1,024 bytes.

It is important that the terminal keyboard is in the CAPS LOCK mode since the program only reads capitalized alphanumeric during interactive input. A carriage return is entered from the keyboard after responding to each prompt. If inappropriate information is entered, or a file with the incorrect format is specified, the program will indicate that an error reading the file has occurred and will prompt the user for the desired information again.

Reading Input Data

The program begins by prompting the user for the names of the input files to be used for the simulation. A prompt is written to the screen asking for each file name beginning with the contaminant data input file. The file name is entered by typing the file path, the filename, and its extension. This allows the program to read the files if they are stored on the computer disk in a separate directory or allows the program to access the files from a floppy diskette. If the files are in the same directory as the program executable file, then no path designation is necessary. The removal device definition file is read next, followed by the time-dependent data file.

After each file has been read successfully, the program writes a message to the screen indicating that the data file has been read. Also, after each file has been read, the program prompts the user for a decision concerning input file data output by writing a prompt to the screen requesting whether the data file that has just been read should be printed. The user should respond to this prompt with a "Y" or an "N." The "Y" response results in a second prompt for the user to designate how the data file should be printed. The choices are the printer (LPT1), the computer screen (CON), or neither (END). This option is available for the user to view the contents of each data file to make sure the data are correct before the program calculation routine begins. Usually, this option is not necessary if the input files have been created carefully. The "N" response causes the program to move to the next input data prompt.

Once data file input is completed, the program prompts the user for the mission duration in hours. This is the length of time that the particular mission being simulated will last. For example, a 7-day Spacelab mission duration is 168 h and the number 168.0 should be entered at this prompt.

Data Output Decisions

After the data files and mission duration have been read by the program, the user is prompted for a decision on calculation data output. The TCCS computer program is capable of producing output data in several formats and writing this data to multiple output devices. The principle output data consist of standard formatted output data controlled by the print switches defined in the device definition data file and plot data output which is specified interactively by the user for each program execution. The program prompts the user for decisions concerning plot data output and output devices in the following order.

Concentration and Device Efficiency Plot Data Output. Concentration data and removal device efficiency data for each basic time increment can be output to two separate data files. The user is prompted by the program for a decision to output only concentration data, only efficiency data, both concentration and efficiency data, or neither. The user selects these options by entering "C," "E," "B," or "N," respectively, at the prompt. Examples of the data output for both concentration and efficiency plot data are shown by tables 8 and 9.

Table 8. Example concentration plot data output.

Initial Time (h)	Final Time (h)	Concentration (mg/m ³)				
		Cont. 1	Cont. 2	Cont. 3	Cont. 4	Cont. n
0.000	1.000	1.189	0.2432E-03	0.1339	0.1899....	0.2022E-1
1.000	12.000	10.16	0.2077E-02	1.464	1.308	0.2201
12.000	24.000	14.72	0.3012E-02	2.655	1.637 ...	0.3972

Toxic Hazard Index Output. The toxic hazard index is a technique used by toxicologists to determine the toxicological acceptability for a mixture of chemical compounds in the atmosphere. It is analogous to an SMAC in some respects. The overall toxic hazard index is the summation of the ratios of each contaminant's atmospheric concentration to its SMAC. Ideally, this summation should be less than one. A more detailed definition of the toxic hazard index is found in appendix B and reference 12.

The user is prompted by the program for a decision to write the toxic hazard index to the standard formatted output device, to the standard output device and a plot data file, or neither. Each option is selected by entering "Y," "P," or "N" at this prompt.

Standard Formatted Output. Standard formatted calculation data produced by the program can be output to a datafile, the printer, the screen, or none of these options. The datafile option requires the user to input the path, filename, and extension of a file that the program will open and write calculation data to during the execution. Again, the path is necessary only if the datafile is not in the same directory as the program executable file. The program checks the file name to determine whether the file already exists. If the file exists, the program indicates this on the screen and asks the user whether it should be overwritten. The user responds to this prompt by entering "Y" or "N." The "Y" response allows overwriting the present contents of the file, and the "N" response causes the program to prompt the user for another file name. The printer selection is made by entering LPT1 at this prompt and results in formatted calculation data output to the print device for the computer. This option is recommended only if output is desired at the end of the program execution since large quantities of paper are produced. The screen option is selected by entering CON and results in the formatted data to be written to the screen. The data scrolls by rapidly when the CON option is selected, and scrolling may be stopped by holding down the control key and pressing the S key (CTRL S). Scrolling is restarted by typing CTRL S again. The END option terminates program execution. The program is designed to ask the user if any other cases will be

Table 9. Example removal device efficiency plot data output.

Initial Time	Final Time	Device	Efficiency for Each Contaminant								
			1	2	3	4	5	6	7	8	n
0.00	1.00	2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.00	1.00	3	0.000	0.000	0.000	0.021	0.125	0.116	1.000	0.000	0.000
0.00	1.00	4	1.000	1.000	0.056	1.000	1.000	1.000	1.000	0.000	0.000
0.00	1.00	5	0.330	1.000	0.950	1.000	1.000	1.000	1.000	1.000	1.000
0.00	1.00	6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.00	1.00	7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.00	1.00	8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.00	1.00	9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.00	1.00	10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.00	1.00	11	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.00	1.00	12	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.00	1.00	13	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.00	1.00	14	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.00	1.00	15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.00	12.00	3	0.000	0.000	0.000	0.021	0.125	0.116	1.000	0.000	0.000
1.00	12.00	4	1.000	1.000	0.003	1.000	1.000	1.000	1.000	0.000	0.000
1.00	12.00	5	0.330	1.000	0.950	1.000	1.000	1.000	1.000	1.000	1.000
1.00	12.00	6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	11	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	12	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	13	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	14	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.00	12.00	15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
12.00	24.00	3	0.000	0.000	0.000	0.021	0.125	0.116	1.000	0.000	0.000
12.00	24.00	4	1.000	1.000	0.002	1.000	1.000	1.000	1.000	0.000	0.000
12.00	24.00	5	0.330	1.000	0.950	1.000	1.000	1.000	1.000	1.000	1.000
12.00	24.00	6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	11	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	12	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	13	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	14	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12.00	24.00	15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

run before terminating execution. The END option causes this prompt to be written to the screen, and the user responds by entering "Y" or "N." The "Y" response reinitializes the program, and the user is prompted for new input data. The "N" response terminates execution completely.

WARNING: Be careful to specify LPT1, CON, or END when making data output decisions to send data to the printer, the screen, or to reset the program. Any other response will cause the data to be written to a file.

Plot Data File Designation. After the output selections have been made, the program prompts the user for file names for each of the plot data files. A separate file is created for concentration plot data, device removal efficiency plot data, and toxic hazard index plot data. Prompts for each file name are written to the screen only if the user has chosen the options specifying this data for the particular program execution run. As with the standard output device, the user responds to this prompt by entering the path, filename, and extension. The path is not necessary if the file is in the same directory as the program executable file. The program checks to see if the file already exists. If this file exists, the program writes a prompt to the screen asking the user if the file should be overwritten. The user must respond to this prompt by entering a "Y" or "N." The "Y" response opens the file and permits overwriting, while the "N" response causes the program to prompt the user for a new file name.

Calculation Routine Execution

After completing the input and output designations, the program begins the calculation routine. During this routine, information will be written to the computer screen depending on the output selections made. Standard formatted output for each time increment will be written to the screen if print switch 8 in the device definition data file is set to 1 and the CON option for output has been selected. However, if print switch 8 is set to 1 and the LPT1 or file option has been selected, only the increment number and the increment starting and ending times will be written to the screen. Similarly, if print switch 8 is set to 0 and any standard output device selection has been made, the program will write only the increment number and the increment starting and ending times to the screen during the calculation routine. At the end of the calculation routine, the CON output option instructs the program to write the final standard formatted output data to the screen, the LPT1 option sends this data to the printer, and the file option sends the data to the designated file.

During execution, the program may also write error messages to the screen. Most error messages relate to output devices and cause the data to not be written to the specified data file. The program indicates the location of the error in the execution and will continue to execute unless the disk capacity has been exceeded. In this case, the execution terminates.

A second type of warning is written to the screen when the calculation for a contaminant does not converge. This warning indicates a serious problem where the concentration and efficiency calculations for a contaminant did not converge to within the specified convergence error within 20 iterations using the basic time increment and 20 iterations using one-tenth the basic time increment. The program will continue to execute; however, the results for that contaminant will be incorrect. This problem is corrected by either using a smaller basic time increment or increasing the convergence error specified in the device definition data file.

After the calculation routine is complete, the program writes a prompt to the screen asking the user if more cases will be run. The user responds to this prompt by entering "Y" or "N." The "Y" response causes the program to loop to the data entry routine to allow the user to specify input files for the next case. The "N" response terminates the program execution, and the DOS prompt will appear on the screen.

Output Data File Manipulation

The program is capable of producing up to four output data files during a single execution. These files contain the standard formatted output, concentration plot data, device removal efficiency plot data, and toxic hazard index plot data. All of these files are in ASCII format and can be loaded into commercially available word processing and spreadsheet programs for editing and analysis. Specific guidelines for manipulating these files are described.

Standard Formatted Output

The data file containing the standard formatted output may be loaded into a word processing program to be included in a report or memorandum. The format may be edited in the program to suit the user's needs. The font size may have to be reduced to fit on 8.5 by 11 inch paper in the portrait orientation. The landscape orientation is recommended. An example of the standard formatted output is shown by figure 5.

Concentration Plot Data

Concentration plot data consists of the beginning and ending time for each time increment and the concentration for each contaminant in the simulation at the end of the time increment. An example of concentration plot data output is shown by table 8. The data are arranged in columns according to the following:

1. The first column contains the increment beginning time in hours (h)
2. The second column contains the increment ending time in hours (h)
3. All columns beyond the first two contain the sequential (contaminant 1 through n) contaminant concentration for each contaminant in milligrams per cubic meter (mg/m³).

These data can be loaded into a spreadsheet program (Lotus 123 is recommended) and the appropriate ranges selected for preparing plots of concentration as a function of time.

Efficiency Plot Data

Removal device efficiency data consist of the beginning and ending time for each increment, the removal device number (2 through 15), and the decimal of each device for each contaminant for that time increment. This data file is arranged as a square matrix in which each time increment has 14 rows as shown by table 9. Each row corresponds to a removal device and contains columns which include the following:

1. The first column contains the increment beginning time in hours (h)

PROGRAM VERSION 8.0 Alpha 04-10-92

5/27/1992 14:51 SAMPLE.DAT PAGE 1
 TIME INCR 1 INITIAL TIME (HRS)= 0.00 FINAL TIME (HRS)= 1.00
 CONT NAME FINAL CABIN MAC EXCEEDS
 NO. CONC (MG/M3) MAC

1	1,1,2-triCl-1,2,2-triflEthane	1.189	383.0	N
2	1,3-Butadiene	0.2432E-03	221.2	N
3	Methane	0.1339	1771.	N
4	2-butanone	0.1899	59.00	N
5	1-benzo[b]pyrrole	0.3853E-02	0.4800	N
6	methyl hydrazine	0.1093E-03	0.8000E-01	N
7	Ammonia	0.9042E-01	17.40	N
8	Carbon monoxide	0.1046	28.60	N
9	Hydrogen	0.2022E-01	247.3	N

GROUP T-VALUES AS SPECIFIED IN NHB 8080.1B APPENDIX D

-01-	-02-	-03-	-04-	-05-	-06-	-07-	-08-	-09-	-10-	-11-	-12-	-13-	-14-	-15-	-16-
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01

OVERALL T-VALUE

0.01

5/27/1992 14:51 SAMPLE.DAT PAGE 2
 TIME INCR 1 INITIAL TIME (HRS)= 0.00 FINAL TIME (HRS)= 1.00
 RATE OF CONTAMINANT REMOVAL-EACH DEVICE (MG/HR) SHEET 1
 NO. NAME CABIN LEAK DEV3 DEV4 DEV5 DEV6 DEV7 DEV8

1	1,1,2-triCl-1,2,2-triflEthane	270.9	0.2785E-01	0.4108E-04	9.087	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
2	1,3-Butadiene	0.5541E-01	0.5655E-05	0.9341E-07	0.1859E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
3	Methane	30.50	0.3089E-02	0.3489E-05	0.5648E-01	0.2458	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
4	2-butanone	43.27	0.4452E-02	1.069	1.463	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
5	1-benzo[b]pyrrole	0.8779	0.9381E-04	0.1328	0.3084E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
6	methyl hydrazine	0.2490E-01	0.2652E-05	0.3483E-02	0.8718E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
7	Ammonia	20.60	0.2827E-02	31.96	0.9293	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
8	Carbon monoxide	23.83	0.2413E-02	0.1521E-05	0.7547E-08	0.2140	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
9	Hydrogen	4.608	0.4666E-03	0.2861E-06	0.1359E-13	0.4138E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00

Figure 5. Example standard formatted data output.

5/27/1992 14:51 SAMPLE.DAT																	
TIME INCR		1	INITIAL TIME (HRS)=		0.00	FINAL TIME (HRS)=		1.00	PAGE 3								
TOTAL CONTAMINANT MASS REMOVED BY EACH DEVICE (MG)																	
NO.		NAME		CABIN	LEAK	DEV3	DEV4	DEV5	DEV6	DEV7	DEV8						
SHEET 1																	
1	1,1,2-triCl-1,2,2-triFlethane		270.9	0.2765E-01	0.4108E-04	9.087	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00						
2	1,3-Butadiene		0.5541E-01	0.5655E-05	0.9341E-07	0.1859E-02	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00						
3	Methane		30.50	0.3089E-02	0.3489E-05	0.5648E-01	0.2458	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00						
4	2-butanone		43.27	0.4452E-02	1.069	1.463	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00						
5	1-benzo[b]pyrrole		0.8779	0.9381E-04	0.1328	0.3084E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00						
6	methyl hydrazine		0.2490E-01	0.2652E-05	0.3483E-02	0.8718E-03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00						
7	Ammonia		20.60	0.2827E-02	31.96	0.9293	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00						
8	Carbon monoxide		23.83	0.2413E-02	0.1521E-05	0.7547E-08	0.2140	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00						
9	Hydrogen		4.608	0.4666E-03	0.2861E-06	0.1359E-13	0.4138E-01	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00						
1																	
5/27/1992 14:51 SAMPLE.DAT																	
TIME INCR		1	INITIAL TIME (HRS)=		0.00	FINAL TIME (HRS)=		1.00	PAGE 4								
DEVICE REMOVAL EFFICIENCY AT END OF TIME INCREMENT (DEC)																	
NO.		NAME		#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15
1	1,1,2-triCl-1,2,2-triFlethane		1.000	0.000	1.000	0.330	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2	1,3-Butadiene		1.000	0.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	Methane		1.000	0.000	0.056	0.950	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	2-butanone		1.000	0.021	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	1-benzo[b]pyrrole		1.000	0.125	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	methyl hydrazine		1.000	0.116	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
7	Ammonia		1.000	1.000	1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
8	Carbon monoxide		1.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
9	Hydrogen		1.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

2. The second column contains the increment ending time in hours (h)
3. The third column contains the removal device number (dimensionless)
4. All columns beyond the first three contain the sequential contaminant removal efficiency for that device for each individual contaminant (contaminant 1 through n).

It must be noted that the cabin, designated as device number 1, is not included since cabin removal efficiency is not relevant. These data can be loaded into a spreadsheet program (Lotus 123 is recommended) and then sorted by the removal device number column and the time increment ending time column to obtain a time-dependent removal device efficiency profile for each removal device. The appropriate data ranges may then be selected for producing data plots.

Toxic Hazard Index Data

Toxic hazard index data consists of the beginning time, ending time, and overall toxic hazard index calculation for each time increment. This data as shown by table 10 is arranged in three columns which include the following:

1. The first column contains the increment beginning time in hours (h)
2. The second column contains the increment ending time in hours (h)
3. The third column contains the overall toxic hazard index (dimensionless).

These data can be loaded into a spreadsheet program (Lotus 123 is recommended) and the second and third columns selected as the plot data ranges.

Table 10. Example toxic hazard index plot data output.

Initial Time	Final Time	Hazard Index
0.00	1.00	0.01
1.00	12.00	0.07
12.00	24.00	0.11

Macintosh Program Version Execution

The Macintosh program version execution is much more simple than the IBM and compatible program version. Execution is started by positioning the cursor over the TCCS computer program application icon, shown in figure 6, and double clicking with the mouse. The program displays the program banner which contains the program name, version number, and version date and then prompts for the input data. At this prompt, a standard Macintosh file location window is displayed, as shown in figure 7, and the appropriate file is selected by highlighting the file name with the cursor and double clicking with the mouse on the open bar to the right. Similar windows appear for selecting the device definition and time-dependent data files.

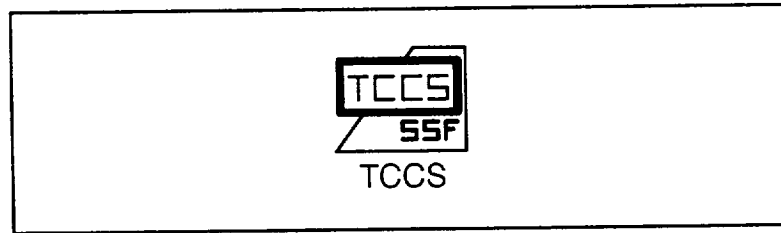


Figure 6. Macintosh program version application icon.

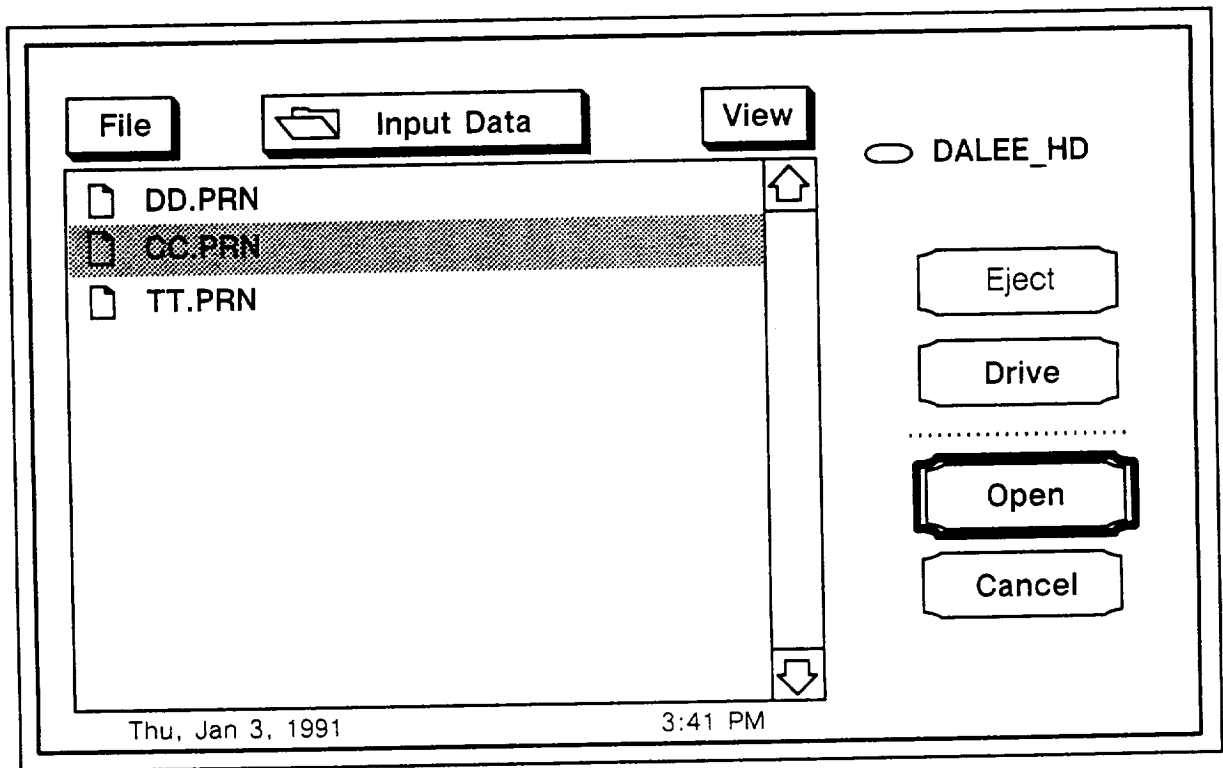


Figure 7. Example Macintosh file selection window.

After the three input files have been located, the main user input screen appears as shown in figure 8. The user enters the mission duration time and selects options to write the runtime data to the screen, files, or by both positioning the cursor in the appropriate box and clicking with the mouse. When the user is finished with this screen, the cursor is positioned over the "OK" box and clicked with the mouse. If any of the data are to be sent to a file, the program prompts for the file name with a standard Macintosh pop-up window similar to the illustration in figure 9.

The final step before the program begins the main calculation routine is to name the plot data output files for concentration and device removal efficiency data. This is accomplished with the standard Macintosh pop-up window shown in figure 9.

TCCS MODEL USER INPUT SCREEN

PRINT CONTAMINANT INPUT DATA?	<input type="checkbox"/> TO SCREEN	<input type="checkbox"/> TO FILE
PRINT DEVICE DEFINITION INPUT DATA?	<input type="checkbox"/> TO SCREEN	<input type="checkbox"/> TO FILE
PRINT TIME DEPENDENT INPUT DATA?	<input type="checkbox"/> TO SCREEN	<input type="checkbox"/> TO FILE
PRINT PROGRAM RUNTIME DATA?	<input type="checkbox"/> TO SCREEN	<input type="checkbox"/> TO FILE

ENTER MISSION DURATION TIME (HRS):

Figure 8. TCCS computer program user input screen.

File
 Plot
View

Conc Plot Data

DALEE_HD

Figure 9. Window for naming plot data output files.

Program solution is initiated after designating the plot files and, upon completion, the user is asked by the program if another run is desired. The user responds to this prompt by typing "yes" or "no." A "yes" response reinitializes the program and prompts the user for new input and output files. A "no" response terminates the program execution and the user may then quit the application or return to the Macintosh Finder. From the Finder, data plotting applications may be selected to analyze the program's concentration and device efficiency data.

DETAILED COMPUTER PROGRAM DESCRIPTION

A detailed description of the TCCS computer program source files is provided to acquaint the user with the main program and each significant subroutine. Flow charts of these routines are provided, and discussion of the theoretical basis for some routines is provided where appropriate. A listing of the program source files is provided in appendix A. This description is paraphrased from a description produced by Lockheed Missiles and Space Company, Inc., under contract NAS8-36406. This work served as the primary reference for this section, and all block flow diagrams were adapted from this document.¹³

Program Editing, Compiling, and Linking

This program was edited, compiled, and linked using the Ryan-McFarland RM/FORTAN® version 2.42 which includes the RM/FORTE® project manager. This FORTRAN compiler is recommended for making changes to the source files.

Main Program

The main program, MAIN, is a simple program with no branching and two loops. A flow diagram is shown in figure 10. Each subroutine required for the particular program run is called during each pass of the main calculation loop until the end of the simulation.

Subroutines CAFILL and RAFILL, which write zeros into all the calculation matrices are called initially to initialize each calculation matrix. Next, CRIN and PRIN are called to read the contaminant, device definition, and time-dependent input data into matrices NN, CDI, DD, and TT. The input data is printed line by line, if desired, by calling subroutines CROUT2 and RROUT2. All initial variables such as time-increment beginning time, time-increment ending time, and the increment counter are zeroed.

The precalculation setup routine, PCSET, is called next. This routine calculates the initial removal efficiency for each removal device, the equilibrium cabin concentration, and the final cabin concentration for a cabin concentration of 1×10^{-20} mg/m³ for all contaminants. Intermediate and final calculation results are stored in matrices CC and DD.

The calculation loop is now entered, and the iterative process of determining the cabin concentration for each contaminant at the end of a time increment is begun. Since cabin concentration is a function of the contaminant mass removed and the contaminant mass removed is a function of the cabin concentration, it is important that the same value be used in the mass balance and removal

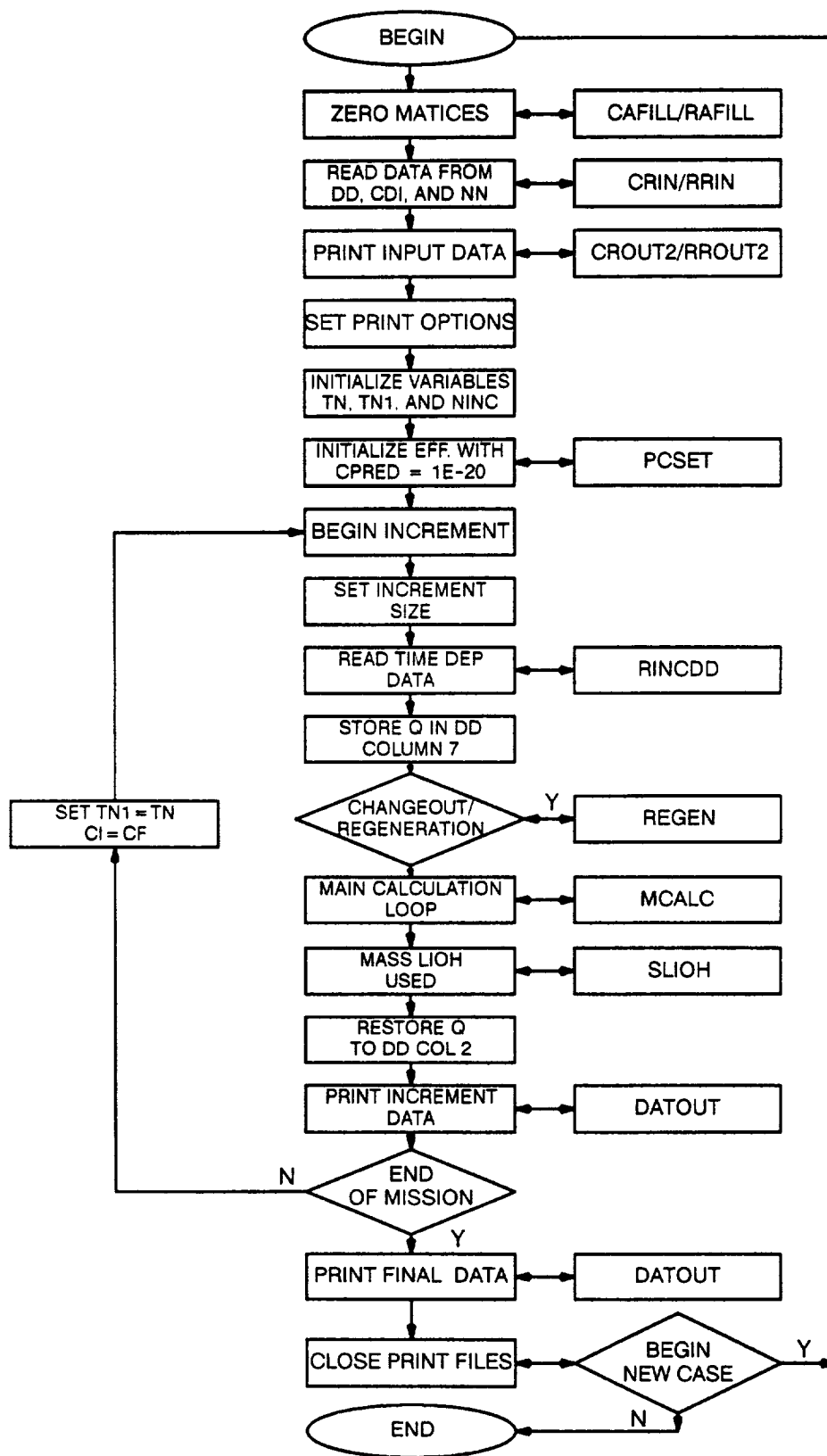


Figure 10. Main program block flow diagram.

efficiency calculation routines. A solution is achieved by assuming an increment concentration, calculating an average contaminant concentration, and comparing the two concentrations. This procedure continues until the assumed and calculated concentrations are equal.

Calculation for each time increment is initiated by increasing the increment counter, setting the increment size, and listing the increment number, starting time, and ending time to the computer screen. Subroutine RINCDD is called to read time-dependent data from matrix TT at this time, and subroutine REGEN is called to check for regeneration of any devices during the time increment and set the adsorbed contaminant masses and device flow rates to zero if necessary. The main calculation loop subroutine, MCALC, is called next to calculate the removal efficiencies, average calculated concentration, and final concentration for each contaminant based on the sum of the mass removed during the previous time increment. Subroutine SLIOH then calculates the amount of LiOH used during the time increment if a LiOH bed is specified in the device definition file. Next, the original device flow rate is restored for any device that was being regenerated during the increment. Subroutine DATOUT is called to print the calculated data at the end of a time increment, if necessary, to both the standard formatted and plot data output devices. The simulation and mission duration times are then compared to determine whether the mission simulation has ended. If the mission simulation has not ended, another pass through the calculation loop begins by setting the new increment beginning time and initial cabin concentration equal to the previous increment ending time and concentration. If the mission simulation has ended, subroutine DATOUT is called to write the final answers to the appropriate output devices as specified by the user. The output files are closed, and the program loops to the beginning to begin another run if the user wishes. If the user has no other runs to make, the program execution is terminated, otherwise, the calculation matrices are zeroed and new input data is supplied to the program for the next run.

Brief descriptions for each major TCCS computer program subroutine are provided in the order that they are called by MAIN. Figure 11 lists the subroutines as they are called and provides a brief description of each subroutine's purpose. Block flow diagrams are provided for the most significant subroutines.

Calculation Loop Subroutines

The following subroutines comprise the principal calculation framework for the TCCS computer program.

Subroutine CAFILL

The subroutine CAFILL is called by MAIN and fills the matrix NN with blanks. Matrix NN contains the contaminant names during the simulation run.

Subroutine RAFILL

The subroutine RAFILL is called by MAIN and fills the matrices CC, TT, CDI, and DD with zeros. Matrices CDI and CC contains contaminant input and calculation data, matrix DD contains device calculation data, and matrix TT contains time-dependent data. This routine is used at the beginning of a computer simulation to initialize these matrices in the event a previous run has been made.

SUBROUTINE LEVEL					DESCRIPTION
1	2	3	4	5	
MAIN	CAFILL RAFILL CRIN RRIN CROUT2 RROUT2 PCSET	PRAFIL CNRSUB	ACHBD RCHBD ALIOH COOXID CATBNR CONDHX		MAIN PROGRAM ZERO MATRIX NN ZERO MATRICES CDI, CC, DD, TT INPUT DATA INTO NN AND CDI INPUT FROM FILE TO DD AND TT PRINT DATA FROM NN AND CDI PRINT DATA FROM DD AND TT PRECALCULATION SETUP FOR ALL CONT ZERO MATRIX DD COLUMNS 17-21 CALCULATE REMOVAL EFFICIENCY AXIAL CHARCOAL BED EFFICIENCY RADIAL CHARCOAL BED EFFICIENCY AXIAL LIOH BED EFFICIENCY CO OXIDIZER EFFICIENCY CATALYTIC OXIDIZER EFFICIENCY CONDENSATE EFFICIENCY CONTAMINANT MATERIAL BALANCE SUM OF MASS REMOVED BY DEVICES LOAD GENERATION INTO DD COL 19 CALCULATE FINAL AND AVERAGE CONC READ INCREMENT DEPENDENT DATA CALCULATE REGERATION/CHANGEOUT MAIN CALCULATION ROUTINE CALCULATE PREDICTED AVERAGE CONC ZERO MATRIX DD COLUMNS 17-21 PUT LAST INCREMENT EFFICIENCY IN DD MASS BALANCE ROUTINE SUM OF MASS REMOVED BY DEVICES LOAD GENERATION INTO DD COL 19 CALCULATE FINAL AND AVERAGE CONC SOLVE FOR NEW REMOVAL EFFICIENCY ZERO MATRIX DD COLUMNS 17-21 CALCULATE REMOVAL EFFICIENCIES AXIAL CHARCOAL BED EFFICIENCY RADIAL CHARCOAL BED EFFICIENCY AXIAL LIOH BED EFFICIENCY CO OXIDIZER EFFICIENCY CATALYTIC OXIDIZER EFFICIENCY CONDENSATE EFFICIENCY CONTAMINANT MATERIAL BALANCE SUM OF MASS REMOVED BY DEVICES LOAD GENERATION INTO DD COL 19 CALCULATE FINAL AND AVERAGE CONC CALCULATE LIOH USED IN INCREMENT PRINT DATA TO THE SPECIFIED DEVICE(S) PRINT CONCENTRATION DATA ANSWERS PRINT DATA HEADINGS PRINT TOXIC HAZARD INDEX ANSWERS
		MASBAL	CALCM LDIGEN PCAVCF		
	RINCDD REGEN MCALC	PREDCT	PRAFIL LODEFF MASBAL	CALCM LDIGEN PCAVCF	
		CONVRG	PRAFIL CNRSUB		
			ACHBD RCHBD ALIOH COOXID CATBNR CONDHX		
			MASBAL	CALCM LDIGEN PCAVCF	
	SLIOH DATOUT	PRFANS	HEADGS		
		GROUP			

Figure 11. TCCS computer program subroutine listing and description.

Subroutine PCSET

The subroutine PCSET is the precalculation setup routine. PCSET gets calculations started by assuming an initial cabin concentration before the program enters the time calculation loop. Figure 12 shows a flow diagram of PCSET. PCSET sets the initial time increment ending time to $1/240$ of the basic time increment specified in the device definition input file. Subroutine PRAFIL is then called, and columns 17 to 21 are zeroed. These columns are used to store the results of subsequent calculations. CNRSUB is called to calculate each device removal efficiency for an assumed initial contaminant concentration of $1 \times 10^{-20} \text{ mg/m}^3$. Contaminant removal rates and predicted, equilibrium, and final cabin concentrations are calculated by subroutine MASBAL. These calculated values are copied from matrix DD to the calculation matrix, CC, and printed out by subroutines CROUT and RROUT if required.

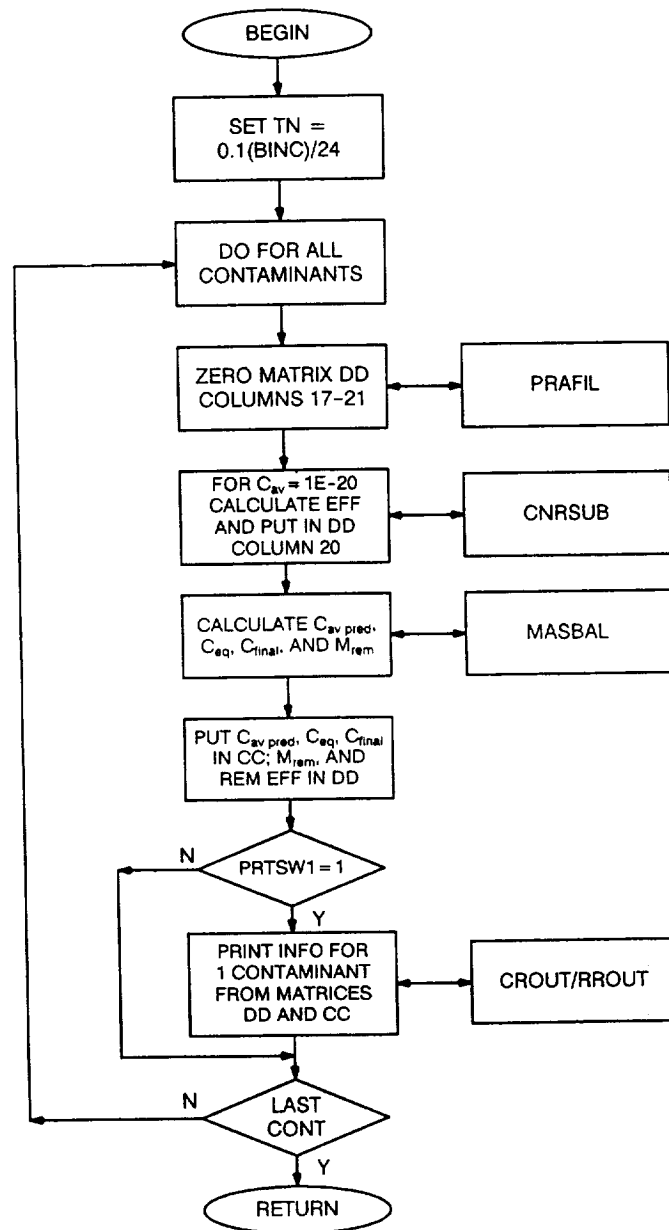


Figure 12. Subroutine PCSET block flow diagram.

Subroutine PRAFIL

Subroutine PRAFIL is called by PCSET and places zeros in matrix DD columns 17 through 21.

Subroutine CNRSUB

The subroutine, CNRSUB, calculates the removal efficiency of each device for each contaminant in the simulation during every time increment. This calculation is based on the average calculated cabin concentration. Figure 13 shows a block flow diagram for CNRSUB. These calculations are conducted by device type rather than the relative positions of each device with respect to each other.

This routine sets the cabin removal efficiency to zero and the leakage efficiency to the maximum of 1.0. Removal efficiencies for all devices with no flow are also set to zero. The remaining device efficiencies are calculated by calling the subroutines ACHBD, RCHBD, ALIOH, COOXID, CATBNR, and CONDHX. These calculated efficiencies are stored in matrix DD.

Subroutine MASBAL

Using the device efficiencies calculated by CNRSUB, MASBAL determines the mass removed, the calculated cabin concentration, the equilibrium cabin concentration, and final cabin concentration for each contaminant during a time increment. This calculation is conducted for all removal devices in parallel and in series. Figure 14 shows a block flow diagram for MASBAL.

MASBAL uses the mass of contaminant removed and the net mass to the cabin to determine the final cabin concentration for each contaminant. The mass of the contaminant removed is defined as the product of the removal device flow rate, contaminant concentration, and device removal efficiency. The net mass of contaminant to the cabin is defined as the difference between the mass generated and mass removed. At steady state or equilibrium, the mass removed equals the mass generated. The mass generated is the sum of all generation sources which include the cabin generation rate and the generation rate in each device. The steady-state of equilibrium concentration is defined according to the following equation:

$$C_{SS} = (m_{\text{net-to-cabin}}) / (\eta_r \times Q) , \quad (2)$$

where $m_{\text{net-to-cabin}}$ is the mass of contaminant, η_r is the overall removal efficiency for all devices, and Q is the atmospheric flow rate through the removal devices.

MASBAL is composed of two parts to determine the steady-state concentration. The first part of MASBAL determines the product of the overall efficiency and flow rate by setting the device generation rates to zero, assuming an arbitrary value for average cabin concentration (100 mg/m³) and cabin generation rate (50 mg/h), and calling CALCM to determine the sum of mass removed for all the removal devices. The second part of MASBAL evaluates the net mass to the cabin by setting the average cabin concentration equal to zero, restoring the contaminant and device generation rates to the values specified in the contaminant data matrix, and calculating the mass removed using CALCM. The $m_{\text{net-to-cabin}}$ equals the difference between the masses generated in the cabin and removal devices and the mass removed. From these values, C_{ss} is calculated according to equation (2). After calculating the steady state concentration, the final and average cabin concentrations are calculated by calling PCAVCF and CALCM is called to calculate the mass removed by the cabin and each device using the average calculated cabin concentration.

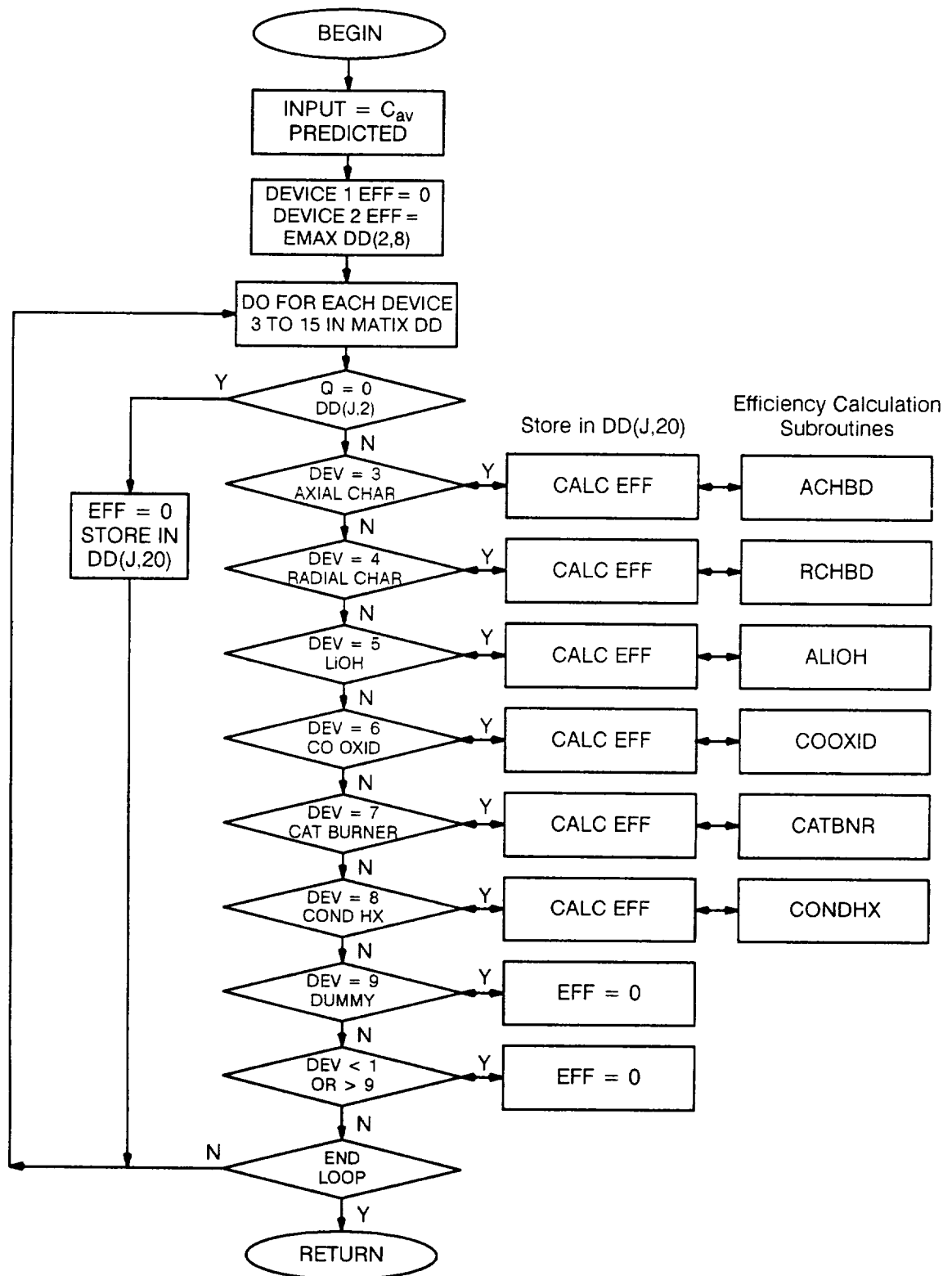


Figure 13. Subroutine CNRSUB block flow diagram.

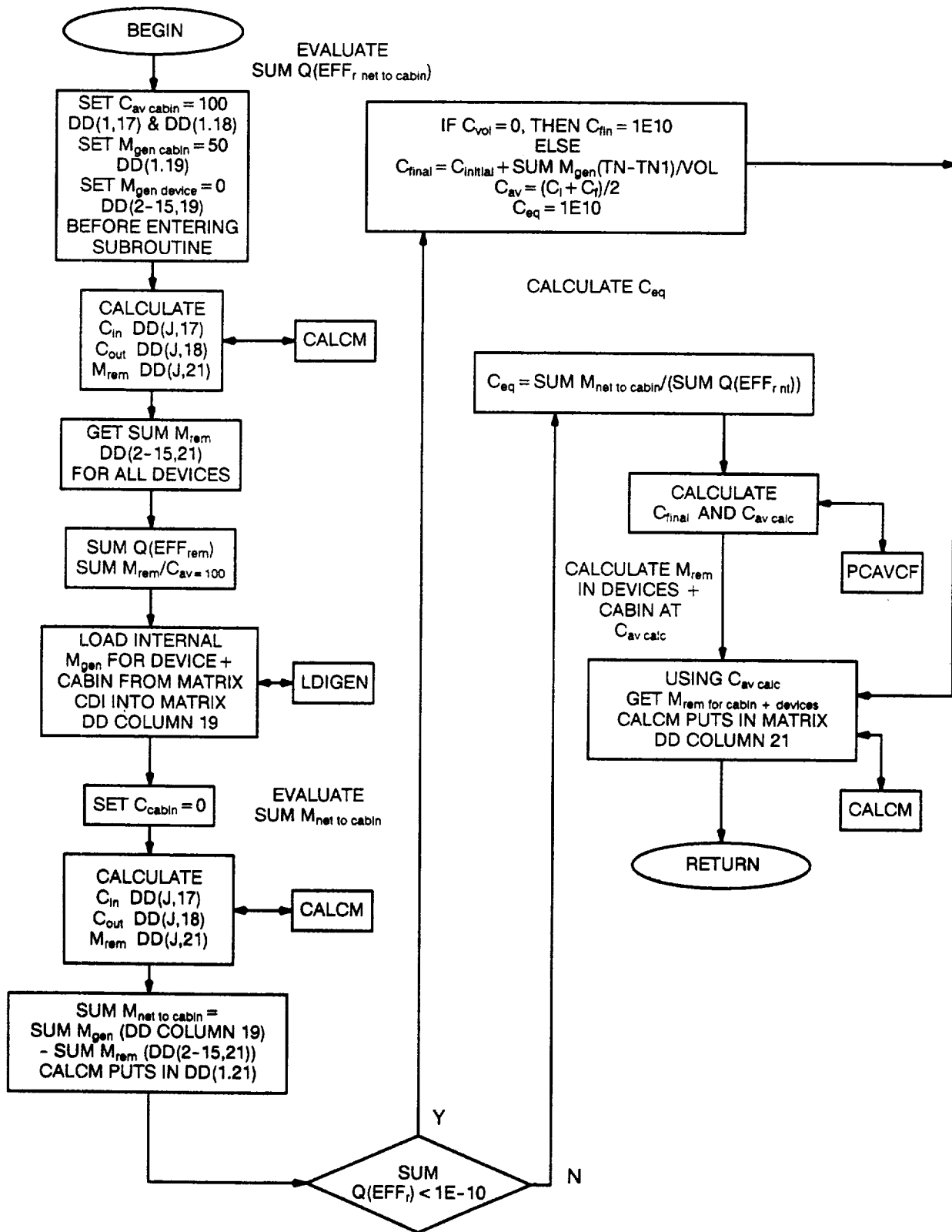


Figure 14. Subroutine MABAL block flow diagram.

Subroutine CALCM

The removal device inlet and outlet concentrations and the total mass removed by the cabin and the specified removal devices is calculated by CALCM by using the removal efficiencies, generation rates, and average cabin concentration. This calculation is sequential from one device to another and uses the outlet concentration of an upstream device as the inlet concentration for a downstream device. This calculation requires the device definition input data to be arranged to allow calculations for all upstream devices to be completed before calculations for the downstream devices. Figure 15 shows a block flow diagram for CALCM.

The subroutine sets the cabin and leakage device inlet and outlet concentrations equal to the average cabin concentration. All other devices are tested for zero flow. Devices with zero flow have their inlet concentration, outlet concentration, and mass removed set equal to zero. Upstream devices for each removal device are identified. If the upstream device type is 1 or 2, the inlet concentration is set equal to the average cabin concentration plus any internal device generation rate divided by the device flow rate. Upstream device types other than 1 or 2 cause the device inlet concentration to be based on the flow rates and outlet concentrations of all the upstream devices.

Outlet concentration and the mass removed by the devices are calculated according to the following equations:

$$C_{out} = C_{in}(1-\eta_r) \quad (3)$$

$$m_{rem} = C_{in}(Q)(\eta_r) \quad (4)$$

The inlet concentration for a device with an upstream device is set equal to the outlet concentration for the upstream device. A device with multiple upstream devices requires the mixing of streams with varying concentrations to be considered. For example, the inlet concentration for a device with three upstream devices must be calculated according to the following equation:

$$C_4 = (C_1 Q_1 + C_2 Q_2 + C_3 Q_3) / Q_4 \quad (5)$$

The sum of the mass removed and mass generated is calculated by adding the masses removed and masses generated by all the devices. The difference between the sum of the mass generated and the sum of the mass removed gives the mass removed by the cabin.

Subroutine LDIGEN

Subroutine LDIGEN is called by MASBAL to load the generation rates from matrix CDI column 1 and columns 10 through 22 into matrix DD column 19.

Subroutine PCAVCF

Subroutine PCAVCF is called by MASBAL to calculate the increment final and average cabin concentrations for each contaminant.

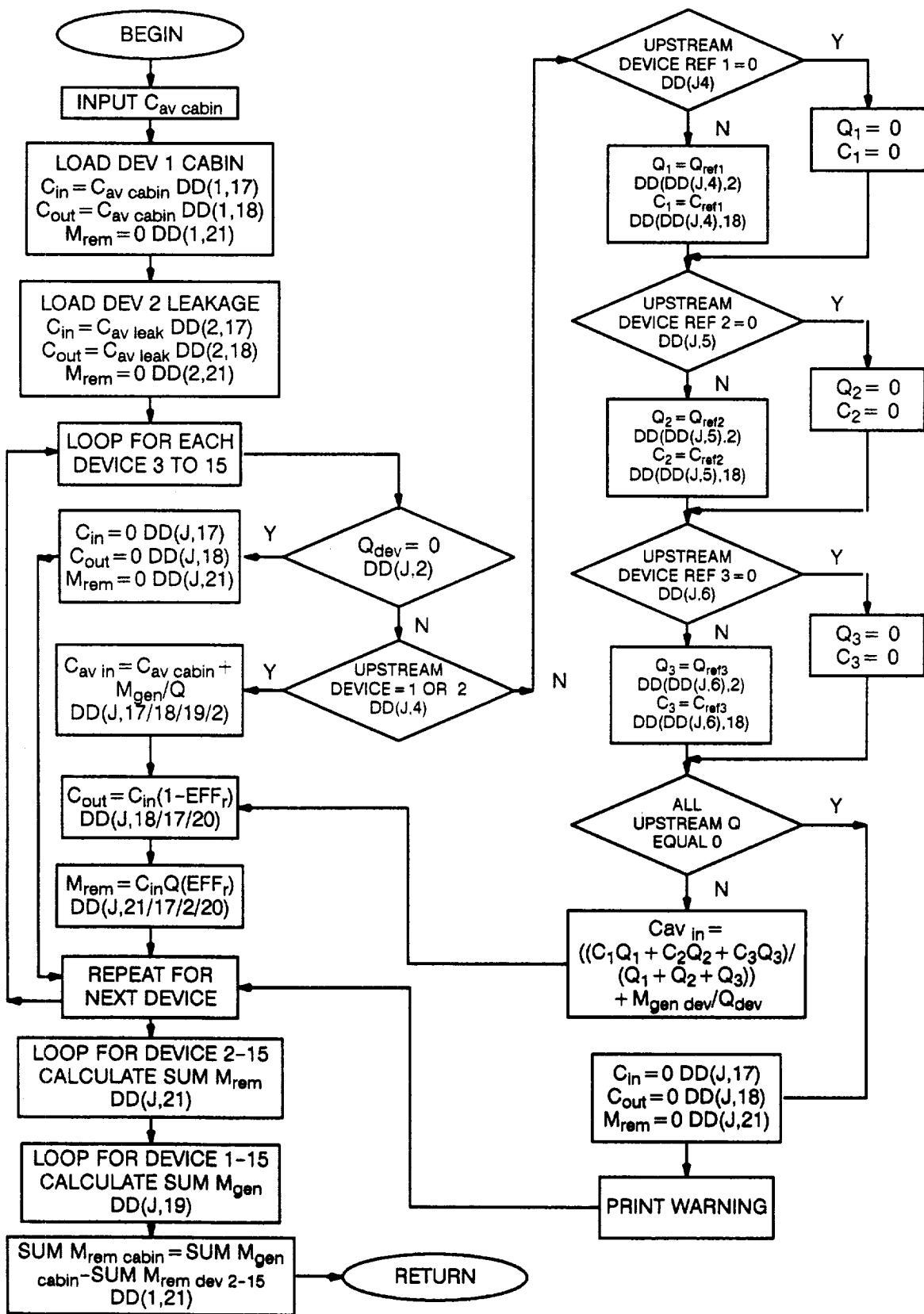


Figure 15. Subroutine CALCM block flow diagram.

Subroutine RINCDD

Subroutine RINCDD is used at the beginning of each time increment to input and operate on the time-dependent data. A flow diagram of RINCDD is shown by figure 16. This subroutine checks the time-dependent data to determine whether any changes occur during the current time increment. Variables in matrix TT are identified. If a contaminant generation rate is indicated, the new rate is placed in the calculation matrix, CDI. Likewise, if a change in removal device flow rate or any other device change is indicated, the new information is placed in the appropriate device definition matrix, DD, location.

Subroutine REGEN

Figure 17 shows a block flow diagram for subroutine REGEN. This subroutine determines whether any charcoal or LiOH beds will be regenerated during the current time increment. If regeneration occurs, the mass of contaminants stored in the beds is set equal to zero. Similarly, if the regeneration duration lasts for the entire time increment, the device flow rate is set equal to zero.

The first check conducted by the subroutine is for device type. Only charcoal and LiOH beds may be regenerated. Only regeneration cases which begin at the time increment beginning or regeneration cases which last for one or more complete time increments are treated. Data concerning the regeneration interval, duration, and first regeneration time are obtained from matrix DD. For the LiOH bed regeneration, the duration is set equal to zero since bed changeout is assumed to occur quickly. The initial time, regeneration time, and regeneration duration are then checked to determine whether they are exact multiples of the basic time increment. If they are not, they are rounded to the next lowest multiple of the time increment and a warning is written to the screen.

The next checks conducted by the routine determine whether regeneration occurs at the beginning of a time increment and whether the regeneration lasts for the entire increment. Regeneration for the entire increment causes the program to deactivate this device for that increment by setting the device flow rate equal to zero. Regeneration at the beginning of an increment causes the sum of the mass removed by that device to be set equal to zero. For an LiOH bed, the total mass of LiOH used is also set equal to zero.

Subroutine MCALC

Calculation of the removal efficiency, mass removed, and calculated, equilibrium, and final cabin concentration for each contaminant and each removal device is controlled by MCALC. These calculations are based on the cumulative mass removed for each contaminant during the previous time increment. Figure 18 shows a block flow diagram for MCALC.

MCALC calls the subroutine PREDCT to calculate the average predicted cabin concentration based on the removal efficiency and the cumulative mass of contaminant removed during the previous time increment and the generation rate during the present time increment. The average predicted concentration is used by the subroutine CONVRG to calculate a new removal efficiency; mass removed; and average calculated, equilibrium, and final cabin concentrations. The predicted and calculated concentrations are compared in CONVRG and recalculated until the difference between them is less than the convergence error specified in the device definition data file, matrix DD. This recalculation and comparison continues for 20 iterations with the full time increment or until the difference is less than the convergence error. If the convergence error is still exceeded after 20 iterations,

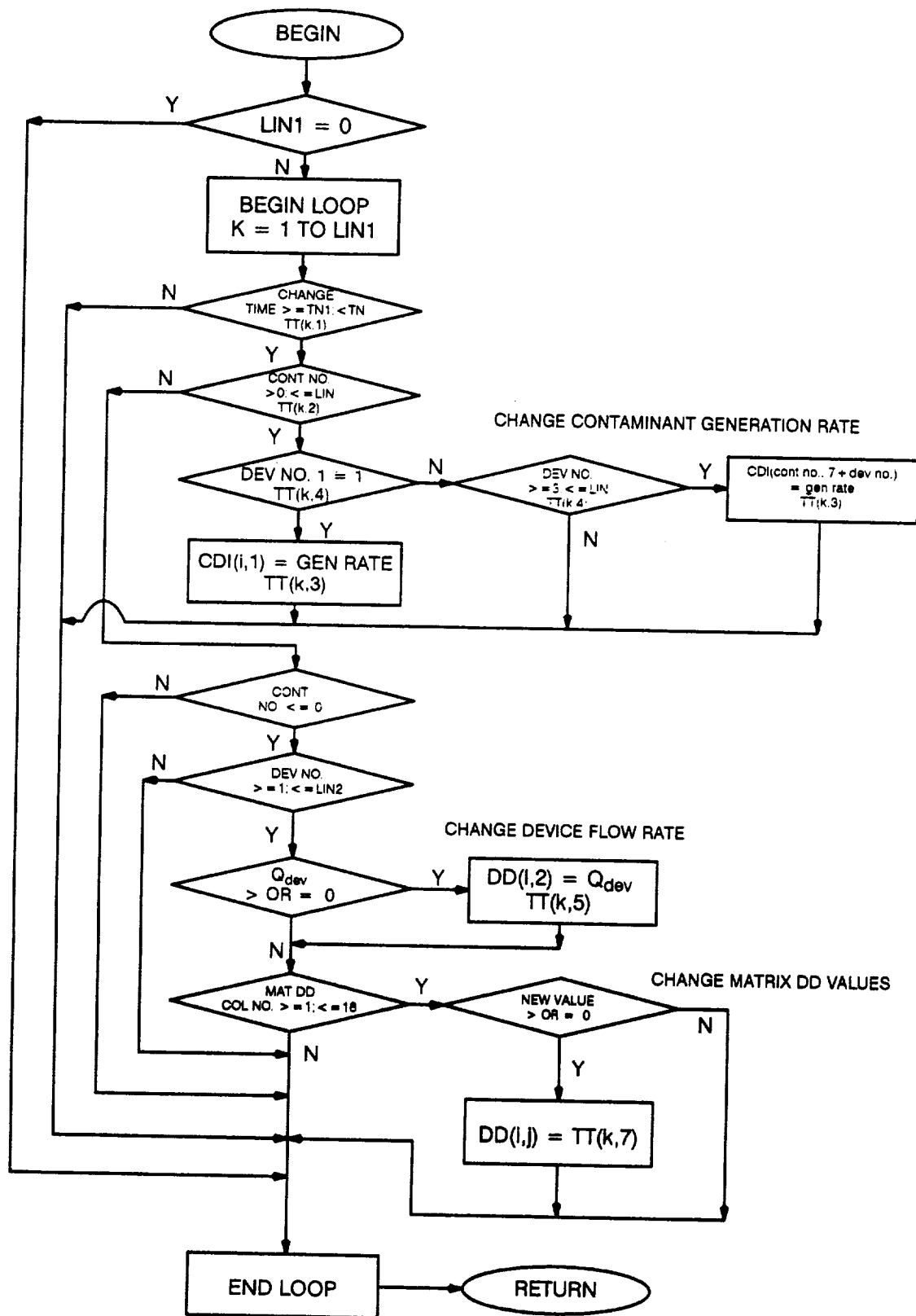


Figure 16. Subroutine RINCDD block flow diagram.

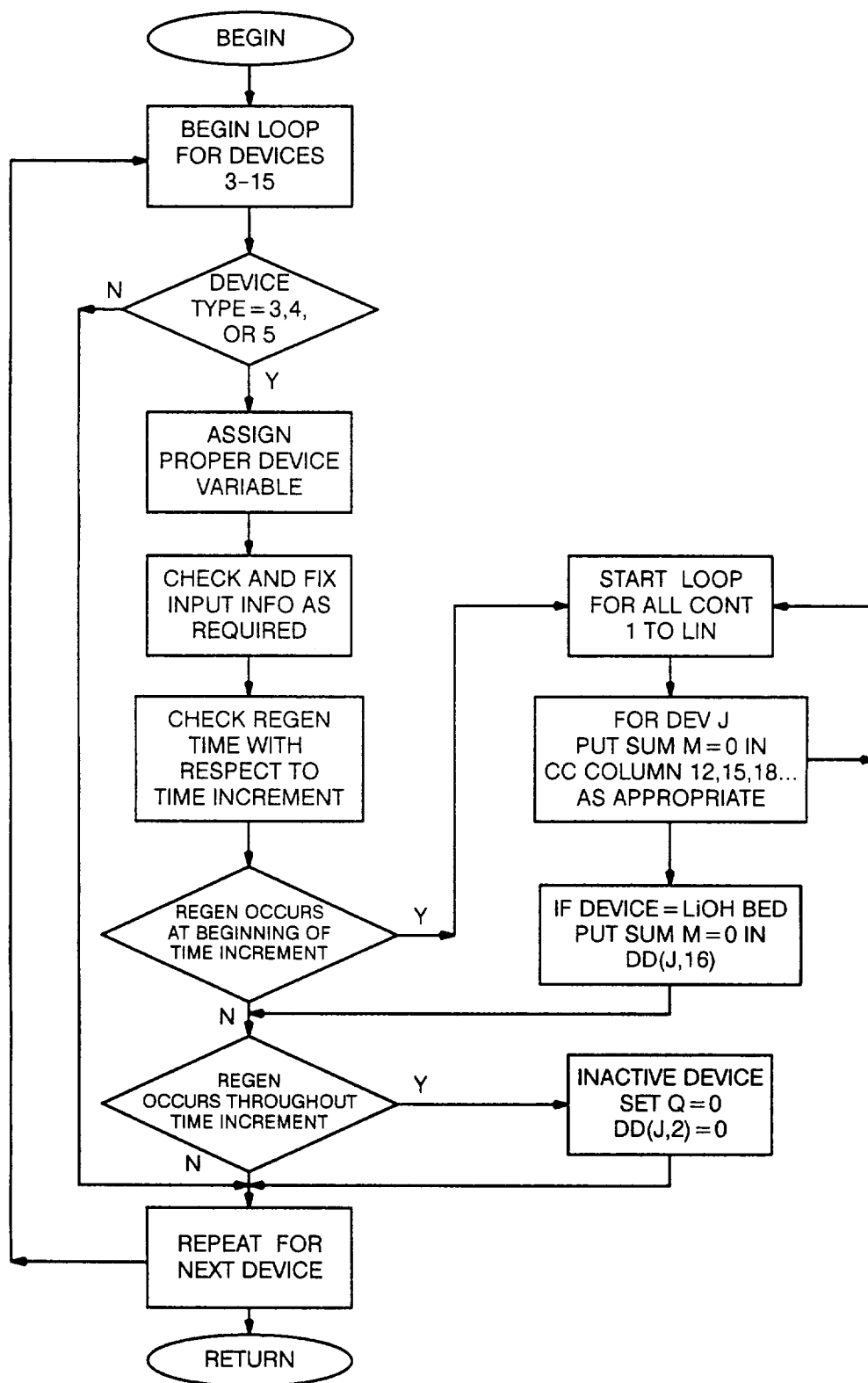


Figure 17. Subroutine REGEN block flow diagram.

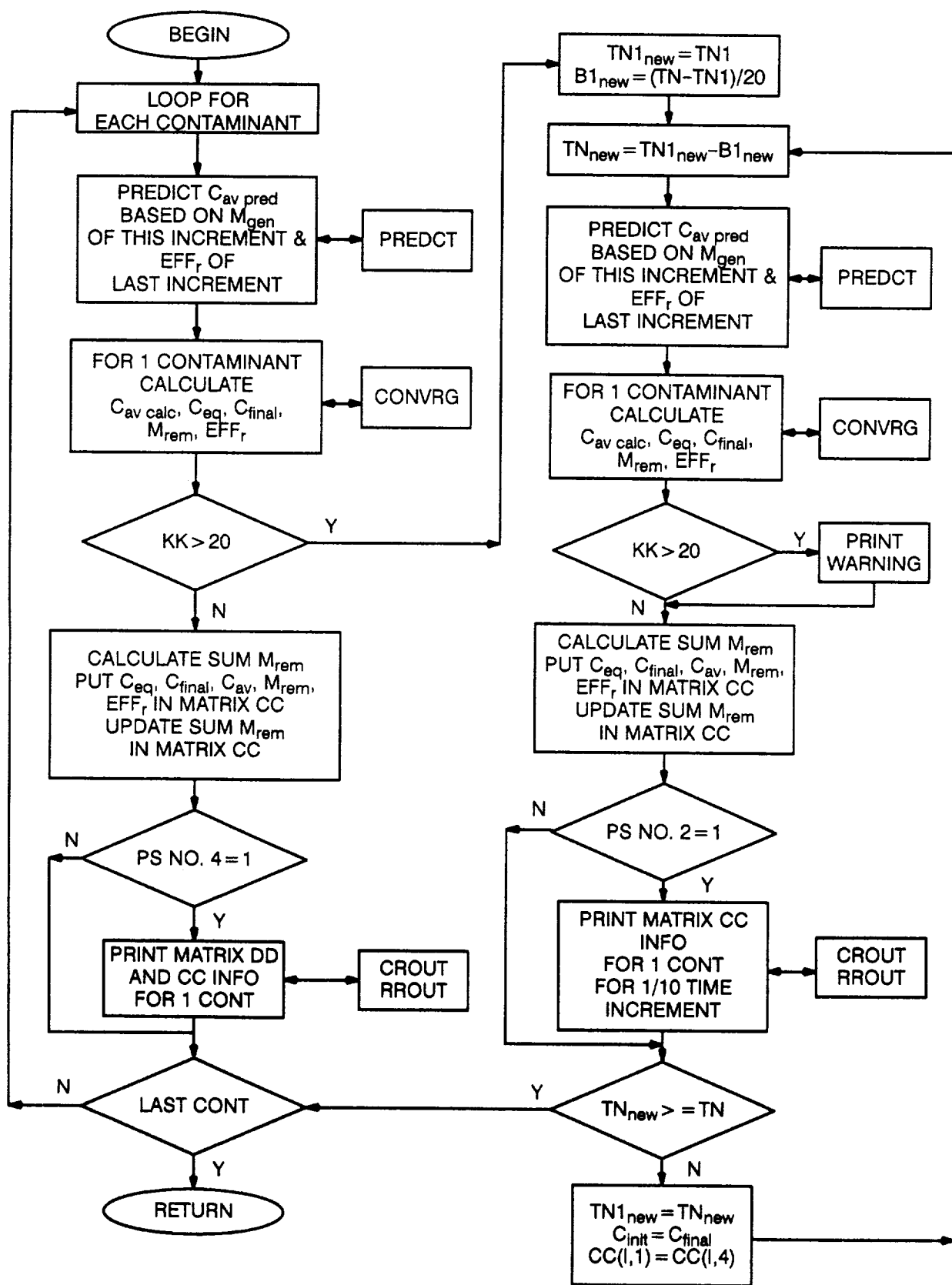


Figure 18. Subroutine MCALC block flow diagram.

another loop with a maximum of 20 iterations is entered which uses one-twentieth the basic time increment for the calculation. This loop ends as soon as the difference between the predicted and calculated values is less than the convergence error or 20 iterations have been completed. If convergence is not attained after this loop, the program writes a warning to the screen indicating that the calculation for the contaminant did not converge. The loop using the one-twentieth time increment is used only for a contaminant that does not converge during the first 20 iterations. This is more efficient than reducing the time increment for all the contaminant calculations.

Subroutine PREDCT

MCALC calls the subroutine PREDCT to calculate the average predicted cabin concentration for each contaminant during each time increment. This calculation is based on the removal efficiency and sum of contaminant mass removed in the previous increment and the generation rate during the present increment. Figure 19 shows a block flow diagram for PREDCT.

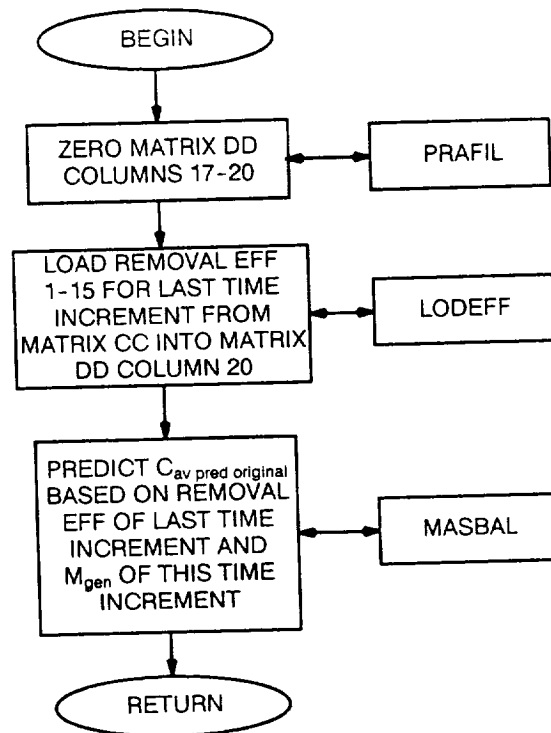


Figure 19. Subroutine PREDCT block flow diagram.

PREDCT calls the subroutine PRAFIL to zero the part of matrix DD required for storing the calculation results. Data from the previous time increment is obtained by LODEFF and MASBAL is called to calculate the efficiency and concentration.

Subroutine LODEFF

Subroutine LODEFF loads the efficiency calculated in the preceding increment from matrix CC to matrix DD.

Subroutine CONVRG

CONVRG is the main convergence loop subroutine. This subroutine calculates the average cabin concentration and compares it with the predicted cabin concentration for each contaminant during every time increment. Figure 20 shows a block flow diagram for CONVRG.

CNRSUB is called by CONVRG to calculate the removal efficiency for each device using the predicted cabin concentration. Based on this removal efficiency, MASBAL calculates the average, final, and equilibrium cabin concentrations and the mass of contaminant removed by each device. The predicted and calculated cabin concentrations are compared by calculating the absolute value of the difference of the predicted and calculated concentration divided by the predicted concentration and comparing that value to the convergence error. If the absolute value of the comparison is less than the convergence error, convergence has been achieved and the iteration stops for that contaminant. If convergence has not been achieved, a new cabin concentration is calculated using a bisection technique after the first iteration and a Newton-Raphson technique for each additional increment. The loop counter value passed back to MCALC which determines whether convergence has been reached within 20 iterations.

Subroutine SLIOH

SLIOH is the subroutine which calculates the cumulative mass of LiOH used during the simulation run. This calculation is cumulative since the mass of LiOH consumed during the present increment is added to the mass consumed in all the previous increments.

Contaminant Removal Device Calculation Subroutines

The contaminant removal device subroutines are supported by a substantial amount of theoretical and experimental data. A brief description of each subroutine is provided in addition to a discussion of the supporting theory and experimental data.

Subroutine ACHBD

Subroutine ACHBD calculates the removal device efficiency for an axial flow charcoal bed. This routine simulates the physical adsorption of contaminants onto the surface of the charcoal. Specially treated charcoals are also considered which include chemical reaction between the surface treatment and the contaminant in addition to adsorption.

A charcoal bed is composed of two zones during the adsorption process. These zones are designated as the saturated zone and the adsorption zone. All contaminant removal takes place in the adsorption zone. The saturated zone provides no net removal since it is in equilibrium with the vapor phase contaminant composition. Figure 21 illustrates the zones simulated by the program graphically. Physical adsorption is an equilibrium process which depends on variables such as the contaminant vapor pressure, inlet concentration, molar volume, and the cabin temperature. Studies conducted by Robell investigated the thermodynamics of adsorption dynamics and developed a correlation between the physical properties of a contaminant and the charcoal saturation capacity. This correlation is based on the Polanyi potential theory and the Gibbs equation. From this study, a correlation factor, called the adsorption potential factor was developed. This factor is defined according to the following equation:

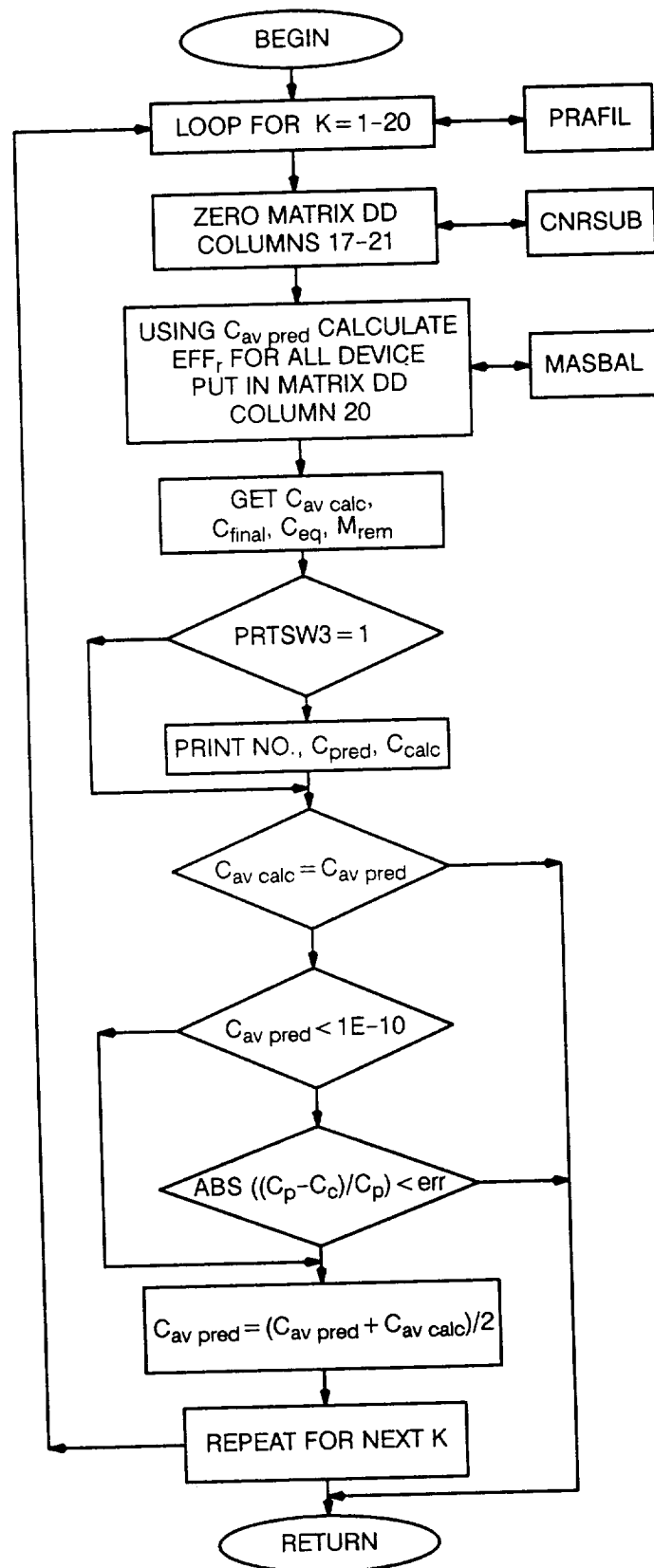
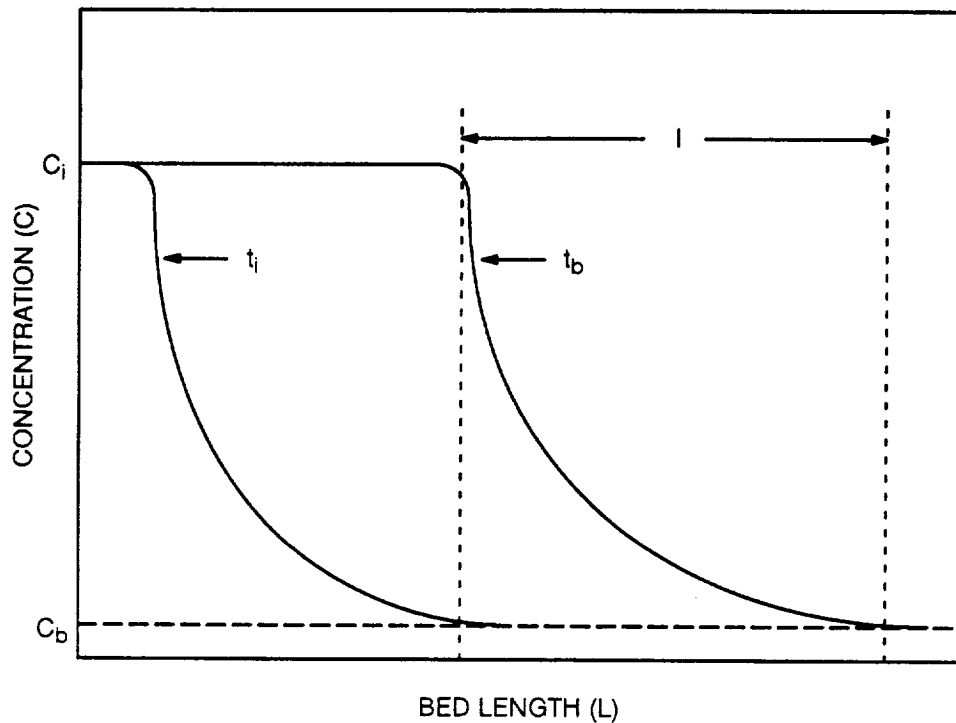


Figure 20. Subroutine CONVRG block flow diagram.



LEGEND

- t_i = Bed profile at the time when a bed segment reaches steady state
- t_b = Bed profile at the service time when the bed outlet concentration equals C_b
- l = Active adsorption zone length
- C_i = Bed inlet concentration
- C_b = Bed penetration concentration

Figure 21. Charcoal saturation and adsorption zone distribution.

$$A = (T/V_m) \log_{10}(p_v/p_c) , \quad (6)$$

where T is the cabin temperature in kelvin, V_m is the contaminant liquid molar volume in $\text{cm}^3/\text{gram-mole}$, p_v is the contaminant vapor pressure at the cabin temperature expressed in concentration units of mg/m^3 , and p_c is the cabin contaminant partial pressure expressed in concentration units of mg/m^3 . This factor was plotted as a function of experimentally determined charcoal saturation capacities to obtain the plot shown by figure 22.¹⁴ The plot in this figure was constructed for Barnebey-Sutcliffe type BD granular activated charcoal. This correlation is not only sensitive to charcoal impregnation and contaminant solubility, but also to relative humidity as shown by figure 23.¹⁵ Additional information may be obtained on charcoal capacity and performance from references 16 and 17. Based on potential plots, empirical equations are obtained which relate the potential factor to the charcoal saturation capacity. The equations used in this program are the following:

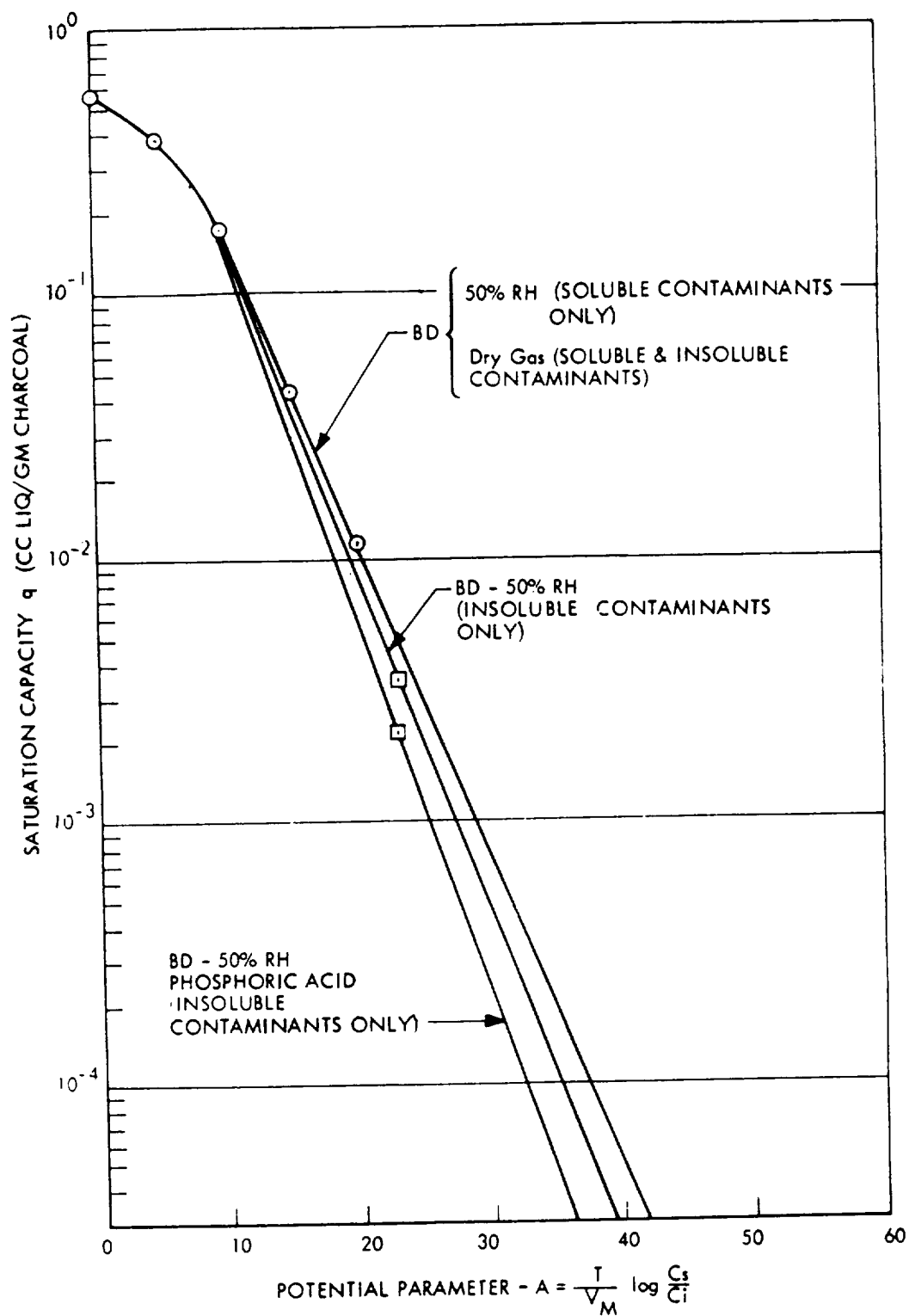


Figure 22. Potential plot for type BD granular activated charcoal.

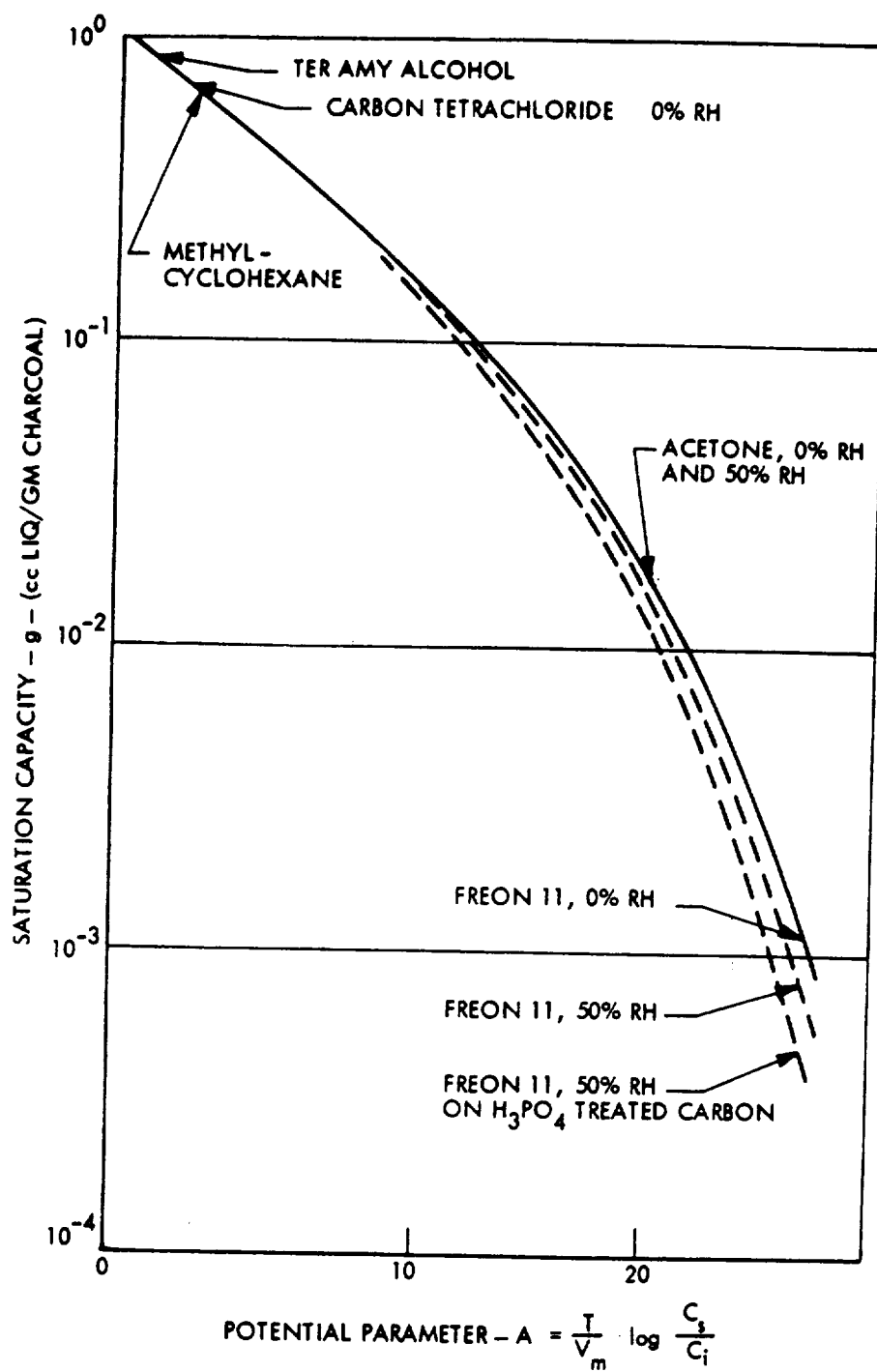


Figure 23. Potential plot showing relative humidity effects.

$$q = 2.2947e^{-0.26492A} \quad (7)$$

$$q = 2.80289e^{-0.28842A} \quad (8)$$

$$q = 3.8636e^{-0.32307A} \quad (9)$$

where q is the charcoal saturation capacity in cm^3 of liquid contaminant per gram of charcoal and A is the adsorption potential factor in kelvin-gmole/cm^3 . As new information concerning adsorption capacity is obtained, these equations can be modified accordingly. In addition, this technique can be applied to other adsorbent materials to simulate other packing materials besides charcoal.

The adsorption zone length for 90-percent removal is determined from experimental data obtained by Olcott at a 0.0066 m/s (1.3 ft/min) flow rate.¹⁸ These data are plotted in figure 24, and the computer program uses the following equation to calculate the adsorption zone length:

$$L_{ads} = (L_{ads} \text{ at } 1.3 \text{ ft/min})(V/1.3)^{0.8} \quad (10)$$

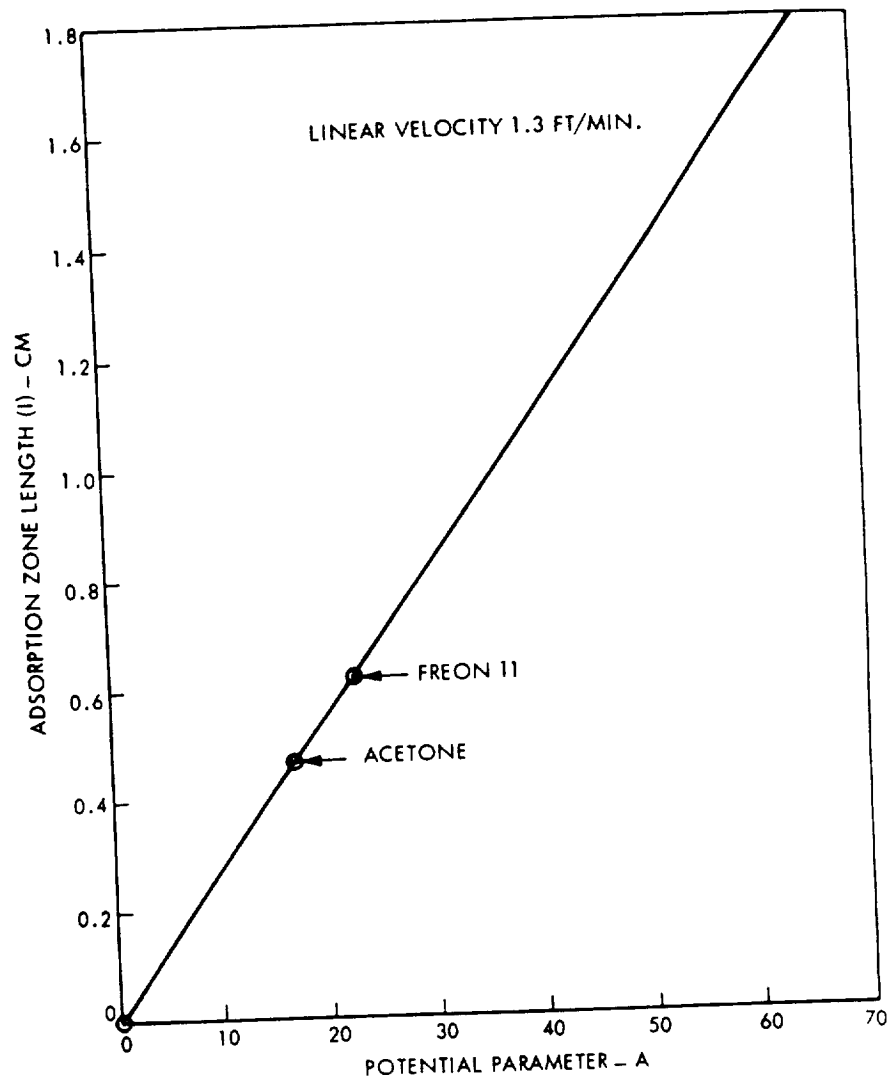


Figure 24. Adsorption zone length as a function of the potential factor.

The adsorption zone length study conducted by Olcott shows the adsorption zone length increases with velocity to the 0.8 power as indicated in the equation. The saturated zone length is based on the charcoal capacity at the prevailing cabin conditions and the amount of contaminant already adsorbed. For a given contaminant mass retained in the bed, the saturation zone length equals the mass of contaminant adsorbed divided by the saturation capacity, q . The total bed length minus the saturated zone length equals the adsorption zone length. The adsorption zone length is the length of the bed actually available for contaminant removal. The program calculates the saturation zone length which is calculated based on the bed geometry, the amount of contaminant removed by the bed, and the saturation capacity, q .

Adsorption of multiple contaminants by charcoal involves some interaction between the contaminants. This interaction, called blockage or coexistence, reduces the capacity of the charcoal to hold other contaminants. Although the theory for coexistence is complex, experimental data indicate that an additional 20 percent can be added to the saturation zone when the calculation for the adsorption zone is conducted.

Reaction with Specially Treated Charcoals. Some contaminants are not readily removed by granular activated charcoal but can be removed by charcoal which has been specially treated with chemicals that react with the contaminant after adsorption onto the charcoal surface. Two commonly used treated charcoals target ammonia and formaldehyde.

Ammonia Removal. Ammonia is removed by treating granular activated charcoal with phosphoric acid. Usually, phosphoric acid loading is two millimoles per gram of charcoal. This results in a requirement of 9.61 grams of charcoal per gram of ammonia to be removed if the reaction goes to completion. The bed removal efficiency is typically 90 to 99 percent per pass for a fresh bed. As the phosphoric acid is depleted, the efficiency drops, eventually reaching zero. This routine assumes the removal efficiency is 100 percent if the bed is less than 80 percent utilized. The efficiency for the last 20 percent of the bed is calculated using the following sine relationship:

$$\eta_r = \sin(m_{\text{charcoal}} - m_{\text{treated charcoal used}})/(0.2)m_{\text{charcoal}} \quad (11)$$

where m_{charcoal} is the mass of charcoal in the bed and $m_{\text{treated charcoal used}}$ is the mass of treated charcoal used.

Formaldehyde Removal. Formaldehyde is removed most efficiently by chromate impregnated charcoal. Manufacturer's data indicate that this charcoal can chemisorb a total amount of formaldehyde equivalent to 5 percent of its weight.¹⁹ Testing at Lockheed Missiles and Space Company, Inc., showed this material's efficiency to drop linearly from 100 percent to 90 percent for an amount of formaldehyde chemisorbed from zero to 0.12 percent of the bed weight. Also, if the bed residence time is less than 0.25 s, the removal efficiency drops linearly.

Subroutine RCHBD

The subroutine, RCHBD, uses the same logic as ACHBD for simulating charcoal adsorption. However, this routine accommodates the geometry of a radial flow charcoal bed.

Subroutine ALIOH

ALIOH simulates removal of acidic contaminants by granular lithium hydroxide and lithium carbonate. The amount of lithium hydroxide consumed per weight of contaminant is calculated from the reaction stoichiometry. This number is input with the contaminant data. Reaction of lithium hydroxide with carbon dioxide to produce lithium carbonate has no noticeable effect on the contaminant removal. The removal efficiency for this device is 100 percent unless the bed is less than 1.905 cm thick or more than 80 percent utilized. The drop in efficiency as the bed is utilized is approximated by the following sine relationship:

$$\eta_r = \sin(m_{\text{LiOH}} - m_{\text{LiOH used}})/(0.2)m_{\text{LiOH}} \quad (12)$$

where m_{LiOH} is the mass of the lithium hydroxide bed and $m_{\text{LiOH used}}$ is the mass of lithium hydroxide utilized. Efficiency for a bed less than 1.905 cm thick decreases linearly with thickness.

Subroutine COOXID

Ambient temperature catalytic oxidation of carbon monoxide and hydrogen is simulated by the COOXID subroutine. This routine simulates ambient temperature catalytic oxidation using a granular activated charcoal with two weight percent platinum loading. This simulation is effective only for carbon monoxide and hydrogen. Efficiency remains constant at 100 percent per pass unless the residence time falls below 0.2 s. For residence times below 0.2 s, the efficiency decreases linearly according to figure 25.²⁰

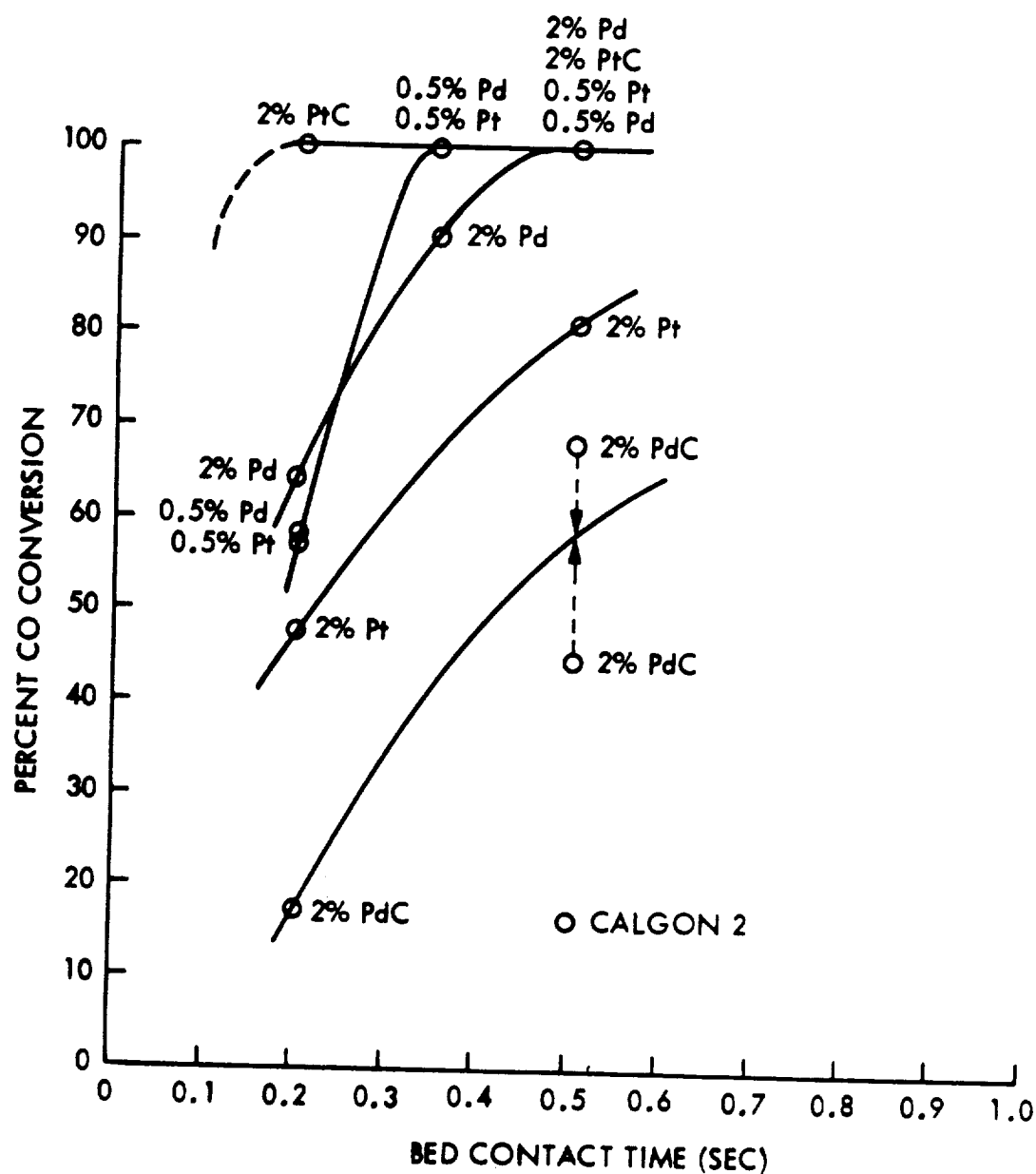
Subroutine CATBNR

The CATBNR subroutine simulates the destruction of hydrogen, carbon monoxide, methane, and other low molecular weight organic contaminants into carbon dioxide and water vapor using high temperature catalytic oxidation. The degree of oxidation in the oxidizer must be input by the user in the contaminant data input file. Typically, oxidation efficiency is based on experimental oxidation performance testing. On average, operating the oxidizer at 400 °C (750 °F) provides removal efficiency of 100 percent for most contaminants.

Subroutine CONDHX

The subroutine, CONDHX, simulates the removal of contaminants by absorption into humidity condensate in a condensing heat exchanger. Some contaminants are removed by this route by not only absorption but also chemical reaction in the condensate. Ammonia is treated in this manner since it dissociates in water and reacts with dissolved carbon dioxide.²¹ All other contaminant removal is simulated using Henry's Law. Using Henry's Law is justified for trace contaminants since their concentrations in the atmosphere approach infinite dilution. Henry's Law correlates the concentration of a contaminant in the atmosphere to its concentration in the liquid phase. The correlation coefficient is the Henry's Law Constant, H , which has units of atmospheres/mole fraction. Equation (13) shows the Henry's Law relationship in which p_c is the contaminant partial pressure in atmospheres, H is the Henry's Law constant in atmospheres per mole fraction, and x is the liquid phase mole fraction.

$$p_c = Hx \quad (13)$$



NOTES:

1. ALL NOBLE METALS ON ALUMINA SUBSTRATE
UNLESS OTHERWISE STATED
2. 2% PtC = 2% PLATINUM ON CARBON
3. 2% PdC = 2% PALLADIUM ON CARBON

Figure 25. Noble metal CO catalyst performance.

The simulation assumes that the absorption process is cocurrent and that equilibrium is very closely approached. A material balance on this process provides a relationship for the condensate mole fraction shown by:

$$x = y/(C/A + H/P) . \quad (14)$$

In this equation, x is the liquid phase mole fraction, y is the vapor phase mole fraction, C is the condensate mass molar flow rate in moles/h, A is the atmospheric molar flow rate in moles/h, H is the Henry's Law constant in atmospheres, and P is the total pressure in atmospheres. Figure 26 illustrates the absorption process. Based on the cabin concentration, the program calculates the inlet mole fraction based on a 1 atmosphere total pressure. The condensate flow rate and atmosphere flow rate are entered in the device definition data and converted to molar flow rates based on 1 atmosphere pressure and 294 K absolute temperature. The mole fraction of contaminant leaving in the condensate is used to determine the mass of contaminant removed. The removal efficiency is calculated from the ratio of the difference in mass of contaminant entering and mass of contaminant removed to the mass of contaminant entering.

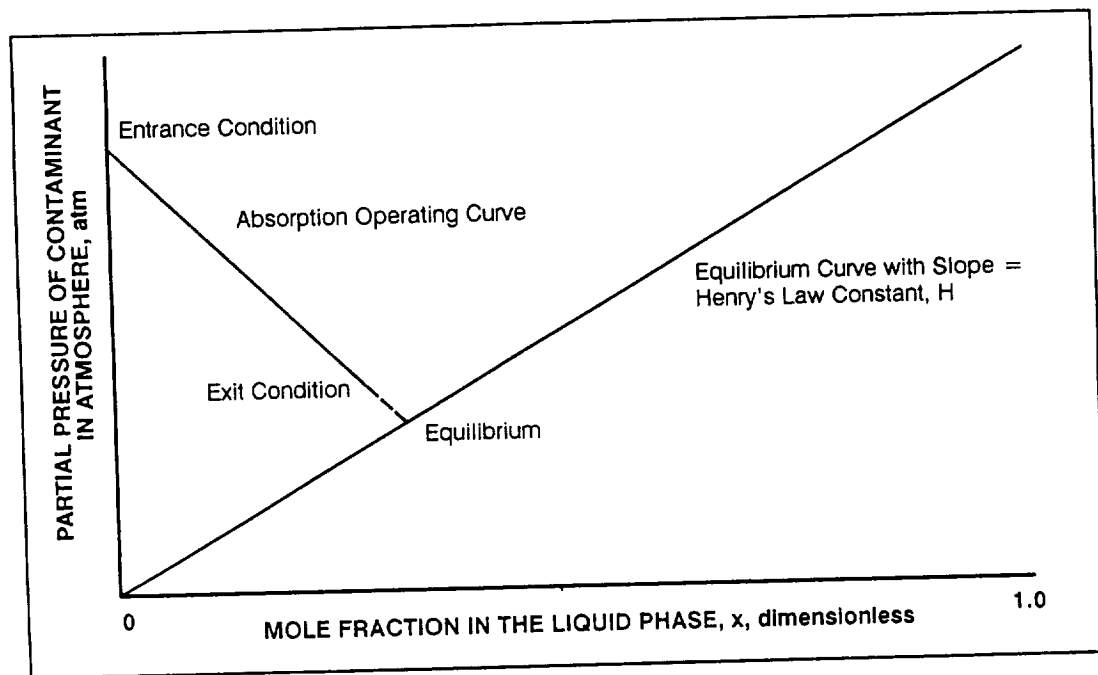


Figure 26. Absorption of contaminants by humidity condensate.

Ammonia removal is treated separately since it reacts chemically with dissolved carbon dioxide in the humidity condensate. According to reference 22, data correlating ammonia partial pressure to liquid phase ammonia concentration for several carbon dioxide atmospheric partial pressures were used to obtain an equation relating liquid and gas phase ammonia composition. These data were obtained by sparging and gas mixture through a volume of water. Figure 27 shows a plot of the result.²² The carbon dioxide curve corresponding to 666.6 Pa (5 mm Hg) was used to obtain equation (15) which relates ammonia mass per kilogram of condensate to the entering ammonia concentration.

$$m_a = 189.6C_c^{0.535} . \quad (15)$$

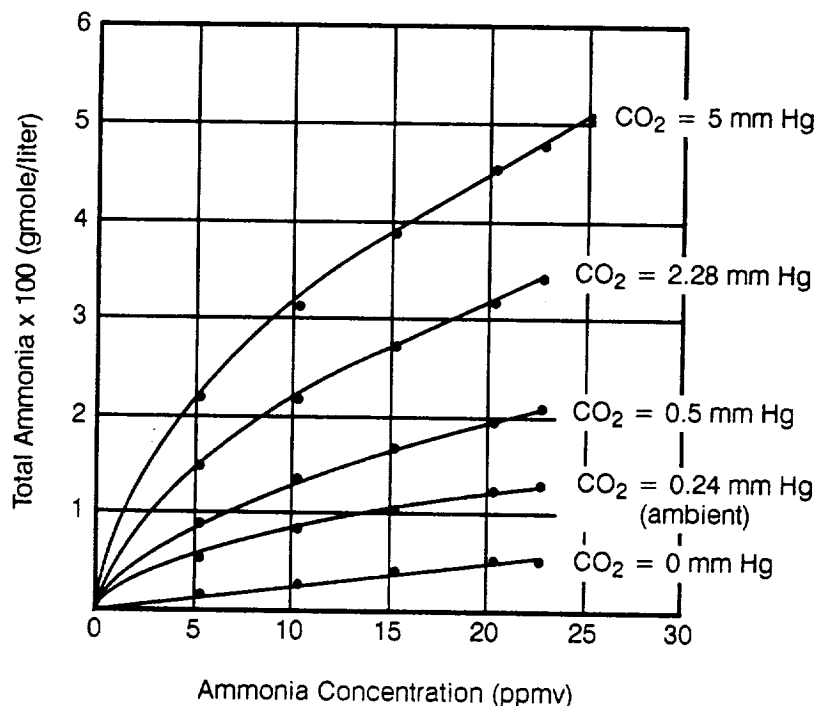


Figure 27. Ammonia solubility in water at varying carbon dioxide concentration.

In this equation, m_a is the mass of ammonia in milligrams per kilogram of condensate and C_c is heat exchanger inlet ammonia concentration in mg/m^3 . This equation is used to determine the ammonia removal efficiency from the mass of ammonia entering and leaving the condensing heat exchanger assembly.

Data Input and Output Subroutines

The following subroutines regulate the data input and output for each computer simulation run. These subroutines have been designed to allow maximum flexibility for calculated data output to facilitate data analysis and reporting.

Subroutine CRIN

Subroutine CRIN is called by MAIN and reads contaminant input data into the contaminant name matrix, NN, and the main calculation matrix, CDI.

Subroutine RRIN

Subroutine RRIN is called by MAIN and reads device definition data and time-dependent data into matrices DD and TT, respectively.

Subroutine CROUT2

CROUT2 is called by MAIN and controls output of the contaminant input data to the printer or computer terminal screen. One row at a time without headings is written to these output devices for the user to review before entering the calculation loop.

Subroutine RROUT2

Subroutine RROUT2 is called by MAIN and controls output of the device definition data and time-dependent data to the printer or computer terminal screen. One row at a time without headings is written to these output devices for the user to review before entering the calculation loop.

Subroutine CROUT

CROUT is called by MAIN and regulates output of matrix CC data during each time increment for diagnostic purposes. This subroutine is called only when print switch 5 is set equal to 1.

Subroutine RROUT

Subroutine RROUT is called by MAIN and regulates output of matrix DD for diagnostic purposes. This subroutine is called only when print switch 5 is set equal to 1.

Subroutine DATOUT

DATOUT is called by MAIN and serves as the master output regulation routine. Routines contained within PRFANS are called from DATOUT according to the print switch designations made by the user.

Subroutine GROUP

Subroutine GROUP is called by DATOUT and calculates the toxic hazard index. This subroutine also regulates the output for the toxic hazard index for both the standard formatted output and the plot data output.

Subroutine PRFANS

PRFANS contains several subroutines that are called by DATOUT which regulate the output for contaminant concentration data, sum of contaminant masses removed data, and removal device efficiency data. This subroutine regulates output for both the standard formatted output and the plot data output.

Subroutine HEADGS

Subroutines within HEADGS are called by PRFANS subroutines to regulate standard formatted data output headings. Headings are provided for contaminant concentration data, contaminant removal rate data, sum of contaminant masses removed data, and removal device efficiency data.

APPENDIX A

**TRACE CONTAMINANT CONTROL SIMULATION (TCCS)
COMPUTER PROGRAM
FORTRAN CODE LISTING**

PRECEDING PAGE BLANK NOT FILMED

 62 INTENTIONALLY LEFT BLANK

This appendix contains listings for each major subroutine and the main TCCS computer program. The main program is listed first followed by listings of each subroutine. The subroutines listings are arranged in alphabetical order by name to provide easy reference.

RM/FORTRAN Compiler (V2.42)

Source File: C:\RMFORT\TCC\MAIN.FO Options: /C 80 /L /BY

05/21/92 12:57:09

```

1 C FILE:MAIN08.FOR
2   PROGRAM TCCS08
3 C   *****
4 C   *****
5 C   *           PROGRAM TO MODEL REMOVAL OF SPACECRAFT           *
6 C   *           GASEOUS CONTAMINANTS                             *
7 C   *           VERSION  8.0 Alpha                               *
8 C   *           APRIL  10, 1992                                  *
9 C   *****
10 C  *****
11 C  SUBROUTINES REQUIRED:
12 C    CAFILL-FILL MATRIX WITH ZEROS
13 C    RAFILL-FILL MATRIX WITH ZEROS
14 C    CRIN-READ IN INPUT DATA
15 C    RRIN-READ IN INPUT DATA
16 C    CROUT-PRINT OUT INPUT DATA
17 C    RROUT-PRINT OUT INPUT DATA
18 C    PCSET-PRECALCULATION SET UP ROUTINE
19 C    MCALC-MAIN CALCULATION ROUTINE
20 C    DATOUT-DATA PRINTOUT ROUTINE
21 C    XXXXX-TIME DEPENDENT DATA ROUTINE
22 C    REGEN-REGENERATION OF DEVICES ROUTINE
23 C    SLIOH-SUM LIOH USED IN TIME INCREMENT
24
25 C    NOTE:SUBROUTINES USE ADJUSTABLE SIZE ARRAYS
26 C        WATCH COMPILER OPTIONS/DIMENSIONING IF
27 C        ANY ARRAY IS LARGER THAN 64K BYTES
28 C
29 C    *****DIMENSION MAIN PROGRAM MATRICES*****
30 C    NN=CONTAMINANT NAME MATRIX
31 C    CDI=CONTAMINANT INPUT DATA MATRIX
32 C    CC=CALCULATON MATRIX
33 C    DD=DEVICE DEFINITION MATRIX
34 C    TT=TIME DEPENDENT DATA MATRIX
35 C
36 C    PRESENTLY SET TO HANDLE MAXIMUM OF 150 CONTAMINANTS IN MATRICES
37 C    THIS VALUE =NROW AND IS USED IN ADDRESSING ADJUSTABLE SIZE
38 C    ARRAYS IN SUBROUTINES
39 C    NTTROW IS USED FOR MATRIX TT MAXIMUM LENGTH
40 C
41 C    CHARACTER NN(300)*30
42 C    REAL CDI(300,23)
43 C    REAL CC(300,48)
44 C    REAL DD(15,21)
45 C    REAL DD(15,23)
46 C    REAL TT(300,7)
47

```

```

48
49 C      NOTE:MUST COMPILE SUBROUTINES PROPERLY FOR ADJUSTABLE SIZE
50 C      ARRAYS IF A MAIN MATRIX EXCEEDS 65536 BYTES (REALS=4 BYTES)
51 C
52 C      ***** DECLARE OTHER TERMS USED IN MAIN PROGRAM *****
53 CHARACTER FNAME*24,DES*1,FCPLOT*24,FTPLOT*24,FEPLLOT*24
54 LOGICAL EX
55 INTEGER PRTSW1,PRTSW2,PRTSW3,PRTSW4,PRTSW5,PRTSW6,PRTSW7,PRTSW8,
56 + TVAL,IDEVNO,IDEVN1,IDEVN2,IDEVN3,PRTSW9
57
58 C      ***** PRINT WELCOME AND PROGRAM VERSION NUMBER *****
59 WRITE (*,9)
60 009 FORMAT (1X,'*****'/'
61 +          1X,'          WELCOME TO THE WORLD          '/*'/
62 +          1X,'          OF THE                          '/*'/
63 +          1X,'          SPACECRAFT ATMOSPHERIC TRACE CONTAMINATION '/*'/
64 +          1X,'          CONTROL SIMULATION PROGRAM          '/*'/
65 +          1X,'          -VERSION 8.0 Alpha-                '/*'/
66 +          1X,'          April 10, 1992                      '/*'/
67 +          1X,'*****'/'
68 C
69 C      ***** DEFINE PROGRAM VARIABLES *****
70 C      LIN=NO. OF LINES OF DATA IN MAT NN & MAT CDI & MAT CC
71 C      LIN1=NO. LINES OF DATA IN MAT TT
72 C      LIN2=NO. LINES OF DATA IN MAT DD
73 C      TN=INCREMENT END TIME (HRS)
74 C      TN1=INCREMENT BEGINNING TIME (HRS)
75 C      TMIS=TOTAL MISSION TIME (HRS)
76 C      NINC=NUMBER OF TIME INCREMENTS ELAPSED
77 C
78 C      MAT NN,CC, AND CDI MUST HAVE SAME NO. OF ROWS
79 C      DIMENSIONS OF MAT DD
80      NROW=15
81 C      NCOL=21
82      NCOL=23
83 C      DIMENSIONS OF MAT CC & ROWS IN MAT NN
84      NROW1=300
85      NCOL1=48
86 C      DIMENSIONS OF MAT CDI
87      NROW2=NROW1
88      NCOL2=23
89 C      DIMENSIONS OF MAT TT
90      NTTROW=300
91      NTTCOL=7
92 C      DEVICE NUMBER FOR OUTPUT DATA (SET TO 6 FOR FORM FEED ON OUTPUT)
93      IDEVNO=6
94 C      DEVICE NUMBER FOR MESSAGE OUTPUT
95      IMSGDN=2
96 C      DEVICE NUMBER FOR CONTAMINANT PLOT DATA
97      IDEVN1=10

```

```

98 C      DEVICE NUMBER FOR T-VALUE PLOT DATA
99      IDEVN3=11
100 C     DEVICE NUMBER FOR EFFICIENCY PLOT DATA
101      IDEVN2=12
102 C     *****      END OF DEFINITION SECTION *****
103 C
104 C     ***** ZERO MATRICES*****
105 C     PUT BLANKS IN NAME MATRIX
106 011 CALL CAFILL(NN,1,NROW1)
107 C     PUT ZEROS IN OTHER MATRICES
108      CALL RAFILL(CDI,NROW2,NCOL2)
109      CALL RAFILL(CC,NROW1,NCOL1)
110      CALL RAFILL(DD,NROW,NCOL)
111      CALL RAFILL(TT,NTTROW,NTTCOL)
112 C
113 C     ***** READ IN DATA FROM FILES AND PRINT IT IF DESIRED *****
114 C
115 010 WRITE(*,*)'INPUT CONTAMINANT DATA FILE NAME:  '
116      CALL CRIN(NN,CDI,NROW2,NCOL2,LIN)
117 012 WRITE(*,*)'PRINT CONTAMINANT INPUT DATA? (Y/N) '
118      READ(*, '(A)')DES
119      IF((DES.NE.'Y') .AND. (DES.NE.'N')) GOTO 12
120      IF (DES.EQ.'N') GOTO 20
121      CALL CROUT2(NN,CDI,NROW2,NCOL2,1,NCOL2,LIN,1,LIN,IMSGDN)
122 C
123 020 WRITE(*,*)'INPUT DEVICE DEFINITION TABLE FILE NAME:  '
124 C     NOTE: ONLY 16 COLUMNS ARE IN THE INPUT FILE
125      CALL RRIN(DD,NROW,NCOL,16,LIN2)
126 022 WRITE(*,*)'PRINT DEVICE DEFINITION TABLE? (Y/N) '
127      READ(*, '(A)')DES
128      IF((DES.NE.'Y') .AND. (DES.NE.'N')) GOTO 22
129      IF (DES.EQ.'N') GOTO 30
130      CALL RROUT2(DD,NROW,NCOL,1,16,LIN2,IMSGDN)
131 C
132 030 WRITE(*,*)'INPUT TIME DEPENDENT DATA FILE NAME:  '
133      CALL RRIN(TT,NTTROW,NTTCOL,NTTCOL,LIN1)
134 032 WRITE(*,*)'PRINT TIME DEPENDENT DATA? (Y/N) '
135      READ(*, '(A)')DES
136      IF((DES.NE.'Y') .AND. (DES.NE.'N')) GOTO 32
137      IF (DES.EQ.'N') GOTO 40
138      CALL RROUT2(TT,NTTROW,NTTCOL,1,NTTCOL,LIN1,IMSGDN)
139 C
140 C     *****READ IN MISSION DATA VARIABLES *****
141 C     READ IN MISSION TOTAL TIME (HRS)
142 040 WRITE(*,*)'INPUT TOTAL MISSION TIME IN HOURS:  '
143      READ(*,*) TMIS
144 C
145 C     ***** PRINT SWITCH DEFINITION *****
146 C     1=RESULTS FOR ONE CONTAMINANT IN PCSET
147 C     2=RESULTS FOR 1 CONT & INCR IN 1/20 INCR CONV ROUTINE (IN MCALC)

```

```

148 C      3=CONVERGENCE VALUES IN CONVRG
149 C      4=RESULTS FOR 1 CONT IN MCALC AFTER CAV CALC
150 C      5=MAT CC AND MAT DD AT END OF TIME INCREMENT
151 C      6=PRINT CONC+M.REM+SUM MASS REM+REM EFF(OTHERWISE ONLY CONC DATA)
152 C      7=PRINT OUTPUT WITH NO FORM FEEDS
153 C      8=PRINT ANSWERS DURING EACH ITERATION (IN MAIN PROGRAM)
154 C      9=CONTROL PLOT FILE OUTPUT
155 C      TVAL=CONTROL OUTPUT OF GROUP CONTRIBUTION T-VALUE DATA
156          PRTSW1=NINT(DD(2,9))
157          PRTSW2=NINT(DD(2,10))
158          PRTSW3=NINT(DD(2,11))
159          PRTSW4=NINT(DD(2,12))
160          PRTSW5=NINT(DD(2,13))
161          PRTSW6=NINT(DD(2,14))
162          PRTSW7=NINT(DD(2,15))
163          PRTSW8=NINT(DD(2,16))
164 C
165 C      **** MAKE DECISION ON CONCENTRATION AND EFFICIENCY PLOT DATA ****
166      WRITE (*,*) 'DO YOU WISH TO WRITE INCREMENT DATA TO A PLOT FILE?'
167      WRITE (*,*) '          1. Concentration Data (C)'
168      WRITE (*,*) '          2. Efficiency Data (E)'
169      WRITE (*,*) '          3. Both Concentration and Efficiency Data (B)'
170      WRITE (*,*) '          4. Neither (N)'
171      WRITE (*,*) 'ENTER YOUR SELECTION: '
172      READ (*, '(A)') DES
173      IF (DES.EQ.'C') THEN
174          PRTSW9=1
175      ELSEIF (DES.EQ.'E') THEN
176          PRTSW9=2
177      ELSEIF (DES.EQ.'B') THEN
178          PRTSW9=3
179      ELSE
180          PRTSW9=0
181      ENDIF
182 C
183 C      ***** MAKE DECISION ON T-VALUE OUTPUT *****
184      WRITE (*,*) 'PRINT GROUP CONTRIBUTION T-VALUE DATA?'
185      WRITE (*,*) '          1. Print to Normal Output (Y)'
186      WRITE (*,*) '          2. Print to Normal Output and Plot File (P)'
187      WRITE (*,*) '          3. Do Not Print (N)'
188      WRITE (*,*) 'ENTER YOUR SELECTION: '
189      READ (*, '(A)') DES
190      IF (DES.EQ.'Y') THEN
191          TVAL=1
192      ELSEIF (DES.EQ.'P') THEN
193          TVAL=2
194      ELSE
195          TVAL=3
196      ENDIF
197 C

```

```

198 C      CHANGE TO NO FORM FEED IF PRTSW7=1
199      IF (PRTSW7.EQ.1) THEN
200          IDEVNO=7
201      ENDIF
202
203 C      ***** MAKE DECISIONS ON DATA OUTPUT *****
204 C      THIS IS WHERE ALL PROGRAM OUTPUT DATA FILES ARE OPENED
205 C      THEY MUST BE CLOSED AT THE END OF THE PROGRAM
206
207 C      SECTION WHICH CHECKS FOR EXISTANCE OF OUTPUT FILE & OPENS IT
208 050 WRITE(*,*) ' WRITE OUTPUT TO FILE, PRINTER, SCREEN, OR END?'
209      WRITE (*,*) '          (FILE NAME/LPT1/CON/END) '
210 C      ***** NOTE: LPT1 OUTPUT REQUIRES 132 COLUMNS *****
211      READ(*,'(A)') FNAME
212 C      QUIT IF FNAME=END
213      IF(FNAME.EQ.'END') GOTO 999
214      IF((FNAME.NE.'LPT1').AND.(FNAME.NE.'CON')) THEN
215          INQUIRE(FILE=FNAME,EXIST=EX)
216          IF (EX) THEN
217              WRITE(*,*) 'FILE EXISTS - OVERWRITE? (Y/N) '
218              READ(*,'(A)') DES
219              IF (DES.NE.'Y') THEN
220                  GOTO 50
221              ELSE
222                  OPEN(IDEVNO,FILE=FNAME,STATUS='OLD',IOSTAT=IOVAL)
223              ENDIF
224          ELSE
225              OPEN (IDEVNO,FILE=FNAME,STATUS='NEW',IOSTAT=IOVAL)
226          ENDIF
227      ENDIF
228      IF ((FNAME.EQ.'LPT1').OR.(FNAME.EQ.'CON')) THEN
229          OPEN(IDEVNO,FILE=FNAME,IOSTAT=IOVAL)
230      ENDIF
231      IF(IOVAL.NE.0) THEN
232          CLOSE (IDEVNO)
233          WRITE(*,*) 'IOERROR= ',IOVAL
234          GOTO 50
235      ENDIF
236 C
237 C      ***** OPEN FILE FOR CONCENTRATION PLOT DATA IF PRTSW9=1 OR 3 *****
238 C
239      IF ((PRTSW9.EQ.1).OR.(PRTSW9.EQ.3)) THEN
240 052 WRITE (*,*) 'FILE NAME FOR CONCENTRATION PLOT DATA OUTPUT? '
241          READ (*,'(A)') FCPLT
242          INQUIRE (FILE=FCPLT,EXIST=EX)
243          IF (EX) THEN
244              WRITE (*,*) 'PLOT FILE EXISTS - OVERWRITE? (Y/N) '
245              READ (*,'(A)') DES
246              IF (DES.NE.'Y') THEN
247                  GOTO 52

```

```

248         ELSE
249             OPEN (UNIT=10,FILE=FCPLOT,STATUS='OLD',IOSTAT=IOVAL)
250         ENDIF
251     ENDIF
252     OPEN (UNIT=10,FILE=FCPLOT,STATUS='NEW',IOSTAT=IOVAL)
253 ENDIF
254 IF (IOVAL.NE.0) THEN
255     CLOSE (UNIT=10)
256     WRITE (*,*) 'IOERROR= ',IOVAL
257     GOTO 52
258 ENDIF
259 C ***** OPEN FILE FOR EFFICIENCY PLOT DATA IF PRTSW9=2 OR 3 *****
260 C
261 C IF ((PRTSW9.EQ.2).OR.(PRTSW9.EQ.3)) THEN
262 054 WRITE (*,*) 'FILE NAME FOR EFFICIENCY PLOT DATA OUTPUT? '
263     READ (*,'(A)') FEPLLOT
264     INQUIRE (FILE=FEPLLOT,EXIST=EX)
265     IF (EX) THEN
266         WRITE (*,*) 'PLOT FILE EXISTS - OVERWRITE? (Y/N) '
267         READ (*,'(A)') DES
268         IF (DES.NE.'Y') THEN
269             GOTO 54
270         ELSE
271             OPEN (UNIT=12,FILE=FEPLLOT,STATUS='OLD',IOSTAT=IOVAL)
272         ENDIF
273     ENDIF
274     OPEN (UNIT=12,FILE=FEPLLOT,STATUS='NEW',IOSTAT=IOVAL)
275 ENDIF
276 IF (IOVAL.NE.0) THEN
277     CLOSE (UNIT=12)
278     WRITE (*,*) 'IOERROR= ',IOVAL
279     GOTO 54
280 ENDIF
281 C ***** OPEN FILE FOR T-VALUE PLOT DATA IF TVAL=2 *****
282 C
283 C IF (TVAL.EQ.2) THEN
284 C 056 WRITE (*,*) 'FILE NAME FOR T-VALUE PLOT DATA OUTPUT? '
285     READ (*,'(A)') FTPLOT
286     INQUIRE (FILE=FTPLOT,EXIST=EX)
287     IF (EX) THEN
288         WRITE (*,*) 'PLOT FILE EXISTS - OVERWRITE? (Y/N) '
289         READ (*,'(A)') DES
290         IF (DES.NE.'Y') THEN
291             GOTO 56
292         ELSE
293             OPEN (UNIT=11,FILE=FTPLOT,STATUS='OLD',IOSTAT=IOVAL)
294         ENDIF
295     ENDIF
296 ENDIF
297

```

```

298      OPEN (UNIT=11,FILE=FTPLOT,STATUS='NEW',IOSTAT=IOVAL)
299      ENDIF
300      IF (IOVAL.NE.0) THEN
301          CLOSE (UNIT=11)
302          WRITE (*,*) 'IOERROR= ',IOVAL
303          GOTO 56
304      ENDIF
305 C
306 C
307 C      CALL SYSTEM TIME AND DATE
308 C      THIS MUST BE CALLED ONLY ONCE SO THAT THE TIME AND DATE WILL
309 C      BE THE SAME ON ALL OUTPUT INFORMATION FOR ONE RUN
310      CALL DATTM(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,ISECOND)
311
312 C      SET IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUTS
313      IPGCTR=0
314
315 C      ***** CHECK BASIC TIME INCREMENT *****
316 C      ***** BASIC TIME INCREMENT (HRS)-DD(1,11) *****
317      BINC=DD(1,11)
318 C      ***** TEST FOR BINC=0 (CAUSES ENDLESS TIME LOOP) *****
319      IF (BINC.EQ.0) THEN
320          CLOSE (IDEVNO)
321          WRITE(*,*) ' ERROR-BASIC TIME INCREMENT=0 '
322          GOTO 999
323      ENDIF
324
325 C      ***** ZERO INITIAL VARIABLES *****
326      TN=0
327      TN1=0
328      NINC=0
329 C
330 C      *****
331 C      *      PRECALCULATION SET UP ROUTINE      *
332 C      *****
333 C      FOR ALL CONTAMINANTS ONE AT A TIME AT CAV PRED=1E-20, CALC INIT
334 C      DEV EFF AND LOAD IT INTO MAT CC-ALSO CALC CAVPRD(CAV PREDICTED)
335 C      OUTPUTS TO PRECALC SET UP ROUTINE:
336 C      TN1=INCREMENT INITIAL TIME (HRS)
337 C      BINC=BASIC INCREMENT SIZE (HRS) (REF.=DD(1,11)) PASS IN????
338 C      LIN=NO. OF CONT IN MAT CC AND NN
339 C      DD,NROW,NCOL=NAME & SIZE OF MAT DD
340 C      CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
341 C      CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
342 C      LIN2=NO. DEVICES IN MAT DD
343 C      NN=NAME OF MAT NN
344 C      INPUTS FROM PRECALC SETUP ROUTINE-SUBROUTINE PCSET:
345 C      PUT IN MAT CC
346 C      CAVPRD=PRED CABIN AV CONC (MG/CU M) : =CC(I,2)
347 C      CEQLIB=EQUILIBRIUM CABIN CONT CONC (MG/CU M) : =CC(I,3)

```



```

348 C      CFINAL=FINAL CABIN CONT CONC (MG/CU M):=CC(I,4)
349 C      PUTS REM EFF FROM DD COL 20 IN CC(I,7-10-13-16 ETC)
350 C      PUTS M.REM IN CC(I,5-8-11-14..)
351
352      CALL PCSET(TN1,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,
353 +CDI,NROW2,NCOL2,LIN2,NN,PRTSW1,IMSGDN)
354
355 C      ***** END OF PRECALCULATION SETUP ROUTINE *****
356
357 C      *****
358 C      *      BEGINNING OF CALCULATION FOR EACH TIME INCREMENT      *
359 C      *****
360 100 CONTINUE
361 C
362 C      ***** INCREASE INCREMENT COUNTER *****
363      NINC=NINC+1
364 C      ***** CHECK FOR CHANGES IN BASIC TIME INCREMENT *****
365      DO 105 K=1,LIN1
366          IF (TT(K,1).EQ.TN1) THEN
367              IF ((TT(K,4).EQ.1).AND.(TT(K,6).EQ.11)) THEN
368                  BINC=TT(K,7)
369                  ENDIF
370              ENDIF
371 105 CONTINUE
372 C      ***** SET UP TIME INCREMENT SIZE FOR INCREMENT *****
373      IF (NINC.EQ.1) TN=BINC/24
374      IF (NINC.EQ.2) TN=BINC/2
375      IF (NINC.EQ.3) TN=BINC
376      IF (NINC.GT.3) TN=TN+BINC
377 C      CHECK FOR INCREMENT FINAL TIME > MISSION TIME
378      IF (TN.GT.TMIS) TN=TMIS
379
380 C      ***** LIST INCREMENT NO. AND TIMES TO CONSOLE *****
381      IF ((FNAME.NE.'CON').OR.(PRTSW8.NE.1)) THEN
382          OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
383          WRITE(IMSGDN,65)NINC,TN1,TN
384 065      FORMAT (1X,'INCR NO.= ',I5,' BEGIN & END TIMES (hours)=',
385 +          F8.3,2X,F8.3)
386          CLOSE (IMSGDN)
387      ENDIF
388
389 C      ***** READ TIME DEPENDENT DATA *****
390 C      CALL TIME DEPENDENT DATA SUBROUTINE-RINCDD
391      CALL RINCDD(I,TN,TN1,DD,NROW,NCOL,LIN2,
392 +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,LIN,TT,NTTROW,NTTCOL,LIN1)
393
394 C      *****STORE ORIGINAL Q DEVICE IN DD COL 7 (TAKEN FROM DD COL 2)*****
395      DO 110 J=1,LIN2
396          DD(J,7)=DD(J,2)
397 110 CONTINUE

```

```

398
399 C      ***** CHECK FOR REGENERATION IN TIME INCREMENT *****
400 C      CALL REGENERATION SUBROUTINE REGEN
401 C      CALL REGEN(TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
402 C      +CDI,NROW2,NCOL2,LIN,LIN2,IMSGDN)
403
404 C      ***** CALL MAIN CALCULATION SUBROUTINE *****
405
406 C      OUTPUTS TO MAIN CALC SUBROUTINE-MCALC:
407 C      I=CONTAMINANT NO.
408 C      TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
409 C      DD,NROW,NCOL=NAME & DIM OF MAT DD
410 C      CC,NROW1,NCOL1=NAME & DIM OF MAT CC
411 C      CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
412 C      NN=NAME OF MAT NN
413 C      LIN=NUMBER OF CONTAMINANTS IN MAT NN & CDI
414 C      LIN2=NO. DEVICES IN MAT DD
415 C      INPUTS FROM MAIN CALC ROUTINE-MCALC:
416 C      TO MAT CC
417 C      PUTS CAVCLC,CEQLIV,&CFINAL IN CC(I,2-3 &4)
418 C      PUTS REM EFF FROM DD COL20 IN CC(I,7-10-13 ETC)
419 C      PUTS M.REM FOR EACH DEV FROM DD COL21 IN CC(I,6-9-12 ETC)
420 C      PUTS SUM MASS REM FOR EACH DEV IN CC(I,8-11-14 ETC)
421
422
423 C      CALL MCALC(I,TN,TN1,DD,NROW,NCOL,
424 C      +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,LIN,LIN2,
425 C      +PRTSW2,PRTSW3,PRTSW4,IMSGDN)
426
427 C      ***** CALCULATE LIOH USED IN INCREMENT *****
428 C      CALL LIOH REMOVAL SUBROUTINE SLIOH
429 C      CALL SLIOH(TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
430 C      +CDI,NROW2,NCOL2,LIN,LIN2)
431 C
432 C      ***** RESTORE DEVICE FLOW *****
433 C      RESTORE ORIGINAL DEVICE FLOW RATE FROM DD COL 7 TO DD COL 2
434 C      DO 120 J=1,LIN2
435 C          DD(J,2)=DD(J,7)
436 C      120 CONTINUE
437
438 C      ***** PRINTOUT OF DATA FOR EACH TIME INCREMENT *****
439 C      IF PRTSW5=1 THEN PRINT MAT DD+MAT CC INFO FOR THIS CONTAMINANT
440 C      IF (PRTSW5.EQ.1) THEN
441 C          OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
442 C          WRITE(IMSGDN,*) 'PRINTOUT FOR MAT CC & DD AT END OF TIME INCR'
443 C          WRITE(IMSGDN,*) 'INFO FROM MAT CC'
444 C          CLOSE(IMSGDN)
445 C          CALL CROUT(NN,CC,NROW1,NCOL1,1,NCOL1,LIN,1,LIN,IMSGDN)
446 C          CALL CROUT(NN,CC,NROW1,NCOL1,1,NCOL1,LIN,1,LIN,IMSGDN,NINC,
447 C      + FNAME,IDEVNO,IOVAL)

```

```

448 C      OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
449 C      WRITE(IMSGDN,*)'INFO FROM MAT DD'
450 C      CLOSE(IMSGDN)
451 C      CALL RROUT(DD,NROW,NCOL,1,NCOL,LIN2,IMSGDN)
452      CALL RROUT(DD,NROW,NCOL,1,NCOL,LIN2,IMSGDN,FNAME,IDEVNO,IOVAL)
453      ENDIF
454
455 C      ***** REGULAR PRINTOUT OF DATA FOR EACH INCREMENT *****
456      IF (PRTSW8.EQ.1) THEN
457          CALL DATOUT(TN,TN1,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,
458 +      CDI,NROW2,NCOL2,LIN2,NN,PRTSW6,PRTSW8,PRTSW9,
459 +      IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR,
460 +      TVAL,FCPLOT,IDEVN1,IDEVN3,IDEVN2)
461
462          IF (IOVAL.NE.0) THEN
463              CLOSE (IDEVNO)
464              WRITE(*,*)'PROGRAM DATA OUTPUT ERROR IN INCREMENT = ',NINC
465              GOTO 999
466          ENDIF
467      ENDIF
468 C      ***** CONTROLS PLOT DATA OUTPUT IF PRTSW8=0 *****
469      IF (PRTSW8.EQ.0) THEN
470          IF ((PRTSW9.GT.0).OR.(TVAL.EQ.2)) THEN
471              CALL DATOUT (TN,TN1,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,
472 +      CDI,NROW2,NCOL2,LIN2,NN,PRTSW6,PRTSW8,PRTSW9,
473 +      IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,
474 +      IPGCTR,TVAL,FCPLOT,IDEVN1,IDEVN3,IDEVN2)
475          ENDIF
476      ENDIF
477 C
478 C      ***** CHECK FOR END OF MISSION *****
479      IF(TN.GE.TMIS) THEN
480 C      END TIME LOOP
481      CONTINUE
482      ELSE
483 C      ***** UPDATE FOR NEXT TIME INCREMENT AND REPEAT *****
484 C      SET TFINAL FOR THIS INCR = TINIT FOR NEXT INCR
485      TN1=TN
486 C      SET CFINAL FOR INCR=CINIT FOR NEXT INCR-ALL CONTAMINANTS
487      DO 130 I=1,LIN
488          CC(I,1)=CC(I,4)
489      130 CONTINUE
490      GOTO 100
491      ENDIF
492 C
493 C      *****
494 C      *      END OF CALCULATION FOR EACH TIME INTERVAL      *
495 C      *****
496
497 C      ***** PRINT FINAL ANSWERS AT END OF MISSION IF DESIRED *****

```

```

498 900 CONTINUE
499     IF (PRTSW8.EQ.0) THEN
500     CALL DATOUT(TN,TN1,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,
501 +     CDI,NROW2,NCOL2,LIN2,NN,PRTSW6,PRTSW8,PRTSW9,
502 +     IDEVNO,-1,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR,
503 +     TVAL,FCPLOT,IDEVN1,IDEVN3,IDEVN2)
504
505     IF (IOVAL.NE.0) THEN
506     CLOSE (IDEVNO)
507     WRITE(*,*)'PROGRAM DATA OUTPUT ERROR - FINAL PRINTOUT'
508     GOTO 999
509     ENDIF
510     ENDIF
511 C ***** CLOSE ALL PROGRAM OUTPUT FILES *****
512     CLOSE (IDEVNO)
513     CLOSE (IDEVN1)
514     CLOSE (IDEVN3)
515     CLOSE (IDEVN2)
516 C
517 999 CONTINUE
518     WRITE(*,*)'DO YOU WISH TO RUN ANOTHER CASE? (Y/N)'
519     READ (*,'(A)')DES
520     IF (DES.EQ.'Y') THEN
521     GO TO 011
522     ENDIF
523 C *****
524 C *                               END OF MAIN PROGRAM *
525 C *****
526     END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C      *****
2 C      * FILE:ACHBD.FOR *
3 C      * SUBROUTINE FOR REM EFF-AXIAL FLOW CHARCOAL BED *
4 C      * DOESNT ALLOW FOR DESORPTION *
5 C      *****
6 C
7      SUBROUTINE ACHBD (TN,TN1,CIIN,TCABIN,COEXIS,BEDQ,EMAX,BEDL,
8      +BEDDIA,DENCH,TRTTYP,DCONT,VMOL,MW,VCONC,SOL,SMR,EFF)
9 C      OUTPUT:
10 C      EFF=BED REMOVAL EFF(DEC)
11 C      INPUTS:
12 C      TN,TN1=INCREMENT INITIAL AND FINAL TIMES(HR)
13 C      CIIN=BED INLET CONT CONC (MG/CU M)
14 C      TCABIN=CABIN TEMP (DEG K)
15 C      COEXIS=COEXISTANCE FACTOR
16 C      BEDQ=BED FLOW RATE(CU M/HR)
17 C      EMAX=MAXIMUM BED EFF (DEC)
18 C      BEDL=BED LENGTH (M)
19 C      BEDDIA=BED DIAMETER (M)
20 C      DENCH=DENSITY OF CHARCOAL IN BED (KG/CU M)
21 C      TRTTYP=BED TREATMENT TYPE(1=CI CHAR,2=PHOS ACID, OTHER.#=NONE)
22 C      DCONT=CONT LIQUID DENSITY (GM/CC)
23 C      VMOL=CONT MOLAR VOL(GM/CC)
24 C      MW=CONT MOLECULAR WGT
25 C      VCONC=CONT VAPOR CONCENTRATION AT TCABIN (MG/CU M)
26 C      SOL=HENRY'S LAW CONSTANT FOR WATER SOLUBILITY (ATM/MOLE FRACTION)
27 C      SMR=SUM OF CONT MASS STORED IN BED(MG)-FROM LAST INCR
28 C
29      REAL LPREV,LAVN1,LUTIL,LIMM,LAVAV,LADS,MW
30      INTEGER FACID,FCI
31 C
32 C      SET CIN=CIIN (THIS PREVENTS CIN FROM BEING PASSED BACK UP
33 C      TO OTHER SUBROUTINES IF IT IS SET TO 1E-20)
34      CIN=CIIN
35 C      BED TREATMENT LOGIC
36 C      FACID=FLAG IF BED IS TREATED WITH PHOSPHORIC ACID (Y=1 N=0)
37 C      FCI=FLAG FOR CI CHAR IN BED (REMOVES FORMALDAHYDE)
38      IF (NINT(TRTTYP).EQ.2) THEN
39          FACID=1
40          FCI=0
41      ELSEIF (NINT(TRTTYP).EQ.1) THEN
42          FACID=0
43          FCI=1
44      ELSE
45          FACID=0
46          FCI=0
47      ENDIF

```

```

48 C      TEST FOR NO BED FLOW(BEDQ=<0) OR TN-TN1<=0;BEDL,BEDDIA,DENCH=0
49 C      IF((BEDQ.LE.0).OR.(TN-TN1.LE.0).OR.(BEDL.LE.0).OR.(BEDDIA.LE.0)
50      +.OR.(DENCH.LE.0)) THEN
51          EFF=0
52          GOTO 199
53      ENDIF
54 C      TEST FOR CI CHARCOAL AND FORMALDEHYDE(FCI=1 AND MW=30.03)
55 C      IF((MW.EQ.30.03).AND.(FCI.EQ.1)) THEN
56          CALL CICH(EFF,EMAX,BEDL,BEDDIA,DENCH,SMR,BEDQ)
57          GOTO 199
58      ENDIF
59 C      TEST FOR AMMONIA AND PHOS ACID ON CHAR(FACID=1 AND MW=17.0 )
60 C      IF ((MW.EQ.17.0).AND.(FACID.EQ.1)) THEN
61          CALL ACIDCH(EFF,EMAX,BEDL,BEDDIA,DENCH,SMR)
62          GOTO 199
63      ENDIF
64 C      TEST FOR MOL VOL=0 (NO CHAR REMOVAL)
65 C      IF (VMOL.EQ.0) THEN
66          EFF=0
67          GOTO 199
68      ENDIF
69 C      CHARCOAL REMOVAL EFFICIENCY CALCULATION
70 C      SUPERFICIAL BED VEL(FT/MIN)
71 C      BEDVEL=BEDQ*.06960/BEDDIA**2
72 C      TEST FOR CIN TOO SMALL IN AVAL CALC
73 C      IF (CIN.LT.1E-20) CIN=1E-20
74 C      AVAL=(TCABIN/VMOL)*LOG10(VCONC/CIN)
75 C      ADS ZONE LENGTH FOR 90% REMOVAL (M)
76 C      LADS=AVAL*.000275*(BEDVEL/1.3)**.8
77 C      GET QI(CC LIQ CONT/GM CHAR)
78 C      CALL FQI(AVAL,QI,FACID,SOL)
79 C      LENGTH OF BED PREVIOUSLY USED BY CONT AT THIS C INLET (M)
80 C      LPREV=SMR*1.273E-6*COEXIS/(DCONT*DENCH*BEDDIA**2*QI)
81 C      RATE OF BED USAGE (M BED/ MG CONT)
82 C      LIMM=1.273E-6*COEXIS/(DCONT*DENCH*BEDDIA**2*QI)
83 C      LENGTH OF BED AVAILABLE FOR ADS ZONE AT BEGINNING OF INCR (M)
84 C      LAVN1=BEDL-LPREV
85 C      IF (LAVN1.LT.0) LAVN1=0
86 C      FIX HERE IF DESORPTION IS DESIRED
87 C      IF (LAVN1/LADS.GT.20) THEN
88          EFFAV=EMAX
89      ELSE
90 C          INIT INCR EFF BASED ON C IN AND BED L AVAIL AT BEG OF INCR(DEC)
91 C          EFAVN1=EMAX*(1-EXP(-2.3025851*LAVN1/LADS))
92 C          LOOP FOR EFFICIENCY
93 C          EFFAV=EFAVN1

```

```

98      DO 399 J=1,10,1
99 C      LENGTH OF BED UTILIZ IN INCR (M)
100      LUTIL=CIN*BEDQ*EFFAV*(TN-TN1)*LIMM
101      IF (LUTIL.GT.LAVN1) THEN
102          GOTO 299
103      ELSE
104 C      AVERAGE BED LENGTH AVAIL (M)
105      LAVAV=LAVN1-LUTIL/2
106      IF ((LAVAV/LADS).GE.20) THEN
107          EFFAV=EMAX
108          GOTO 299
109      ELSE
110 C      AV EFF BASED ON AV BED L AVAIL (DEC)
111      EFFAV=EMAX*(1-EXP(-2.3025851*LAVAV/LADS))
112      ENDIF
113      ENDIF
114 399  CONTINUE
115 299  ENDIF
116 C      MAX EFF BASED ON C IN AND RATE OF BED USAGE (DEC)
117      EFFMAX=LAVN1/(CIN*BEDQ*(TN-TN1)*LIMM)
118      IF (EFFAV.GT.EFFMAX) EFFAV=EFFMAX
119      IF (EFFAV.LT.0) EFFAV=0
120      IF (EFFAV.GT.EMAX) EFFAV=EMAX
121 C      EFF=ACTUAL EFF OUTPUT FROM SUBROUTINE
122      EFF=EFFAV
123 C      REMOVE THIS CHECK IF DESORPTION IS ADDED
124 199 IF (EFF.LT.0) EFF=0
125      IF (EFF.GT.EMAX) EFF=EMAX
126      RETURN
127      END

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS IN PROGRAM UNIT: 0

```

128 C      *****
129 C      * SUBROUTINE ACIDCH - CALCULATES REMOVAL EFF *
130 C      * BED WITH NH3 AND .2 MILLIMOLE H3PO4 ON CHAR *
131 C      *****
132      SUBROUTINE ACIDCH(EFF,EMAX,BEDL,BEDDIA,DENCH,SMR)
133 C      OUTPUTS
134 C      EFF=OUTPUT REMOVAL EFF (DEC)
135 C      INPUTS
136 C      EMAX=MAXIMUM BED REMOVAL EFF (DEC)
137 C      BEDL=BED LENGTH (M)
138 C      BEDDIA=BED DIAMETER (M)
139 C      DENCH=CHARCOAL DENSITY(KG/CU M)
140 C      SMR=SUM OF MASS OF CONT REMOVED AT BEG OF INCR (MG)
141 C
142 C      CHAR USED (KG)
143      CHRUSD=9.61E-6*SMR

```

```

144 C      CHAR BED WGT(KG)
145      BEDWGT=BEDL*BEDDIA**2*.785*DENCH
146      IF (CHRUSD.LT.0.8*BEDWGT) THEN
147          EFF=EMAX
148      ELSE
149          EFF=EMAX*SIN((BEDWGT-CHRUSD)*1.57/(BEDWGT*0.2))
150      ENDIF
151 C      PREVENTS NEGATIVE EFF FOR REACTION
152 C      IF (EFF.LT.0) EFF=0
153      IF (EFF.GT.EMAX) EFF=EMAX
154      RETURN
155      END

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS IN PROGRAM UNIT: 0

```

156 C      *****
157 C      * SUBROUTINE CICH      -  CALCULATES REMOVAL EFF      *
158 C      *    FOR FORMALDELYDE AND CI CHAR BED                *
159 C      *****
160      SUBROUTINE CICH(EFF,EMAX,BEDL,BEDDIA,DENCH,SMR,BEDQ)
161      OUTPUTS
162      EFF=OUTPUT REMOVAL EFF (DEC)
163      INPUTS
164      EMAX=MAXIMUM BED REMOVAL EFF (DEC)
165      BEDL=BED LENGTH (M)
166      BEDDIA=BED DIAMETER (M)
167      DENCH=CHARCOAL DENSITY(KG/CU M)
168      SMR=SUM OF MASS OF CONT REMOVED AT BEG OF INCR (MG)
169      BEDQ=BED FLOW RATE (CU M/HR)
170 C
171      BEDWGT=BEDL*BEDDIA**2*.785*DENCH
172 C      PERCENT OF BED WEIGHT CONSUMED (DEC)
173      PBWGT=SMR/(BEDWGT*1E6)
174      IF (PBWGT.LT..0012) THEN
175          EFF=1-PBWGT*83.3
176      ELSE
177          EFF=.9*COS(PBWGT*1.57/.05)
178      ENDIF
179 C      BED RESIDENCE TIME (SEC)
180      BREST=BEDL*BEDDIA**2*3600/(BEDQ*1.273)
181      IF (BREST.LT.0.25) THEN
182          EFF=EFF*BREST/.25
183      ENDIF
184 C      PREVENTS NEGATIVE EFF FOR REACTION
185 C      IF (EFF.LT.0) EFF=0
186      IF (EFF.GT.EMAX) EFF=EMAX
187      RETURN
188      END

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0

NUMBER OF ERRORS IN PROGRAM UNIT: 0

```
189 C *****
190 C * SUBROUTINE FQI - FINDS QI, THE CHARCOAL *
191 C * CAPACITY FOR A CONT (CC LIQ CONT/GM CHAR) *
192 C *****
193 SUBROUTINE FQI(AVAL,QI,FACID,SOL)
194 INTEGER FACID
195 C OUTPUTS
196 C QI=CHARCOAL CAPACITY (CC LIQ CONT/GM CHAR)
197 C INPUTS
198 C AVAL= A VALUE OF CONTAMINANT
199 C FACID= FLAG FOR ACID TREATED CHAR IN BED (Y=1 N=0)
200 C SOL= CONTAMINANT HENRY'S LAW CONSTANT (ATM/MOL FRACTION)
201 C
202 IF (AVAL.LT.0) AVAL=0
203 C A VALUE .GT. 10 AND .LT. 200
204 IF ((AVAL .GT. 10) .AND. (AVAL.LT.200)) THEN
205 C SOLUBLE CONTAMINANT - ACID OR NON-ACID TREATED CHAR
206 IF (SOL.GT.0) THEN
207 QI=2.2947*EXP(-0.26492*AVAL)
208 C INSOLUBLE CONTAMINANT + NON ACID TREATED CHARCOAL
209 ELSEIF (FACID.LE.0.1) THEN
210 QI=2.80289*EXP(-0.28842*AVAL)
211 C INSOLUBLE CONT + ACID TREATED CHARCOAL
212 ELSE
213 QI=3.8636*EXP(-0.32307*AVAL)
214 ENDIF
215 C A VALUE .LE. 10
216 ELSEIF (AVAL.LE.10) THEN
217 QI=.56-AVAL*.04
218 C AVAL .GE. 200
219 ELSE
220 QI=1E-20
221 ENDIF
222 RETURN
223 END
```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0

NUMBER OF ERRORS IN PROGRAM UNIT: 0

NUMBER OF WARNINGS IN COMPILATION : 0

NUMBER OF ERRORS IN COMPILATION : 0

```

1 C      *****
2 C      *          SUBROUTINE  ALIOH  -  AXIAL FLOW BED          *
3 C      *          CALCULATES REMOVAL EFFICIENCY FOR LIOH      *
4 C      *****
5 C
6 C      SUBROUTINE ALIOH(TN,TN1,EMAX,BEDL,DENLI,BEDDIA,RWUTLI,SWUTLI,
7 C      +REMFACT, EFF)
8 C
9 C      INPUTS:
10 C      TN=FINAL INCREMENT TIME (HRS)
11 C      TN1=INITIAL INCREMENT TIME (HRS)
12 C      EMAX=MAXIMUM POSSIBLE REMOVAL EFFICIENCY (DEC)
13 C      BEDL=BED LENGTH (M)
14 C      DENLI=LIOH DENSITY (KG/CU M)
15 C      BEDDIA=BED DIAMETER (M)
16 C      RWUTLI=RATE OF LIOH USAGE FOR ALL CONTAMINANTS FROM LAST INCR(KG/HR)
17 C      SWUTLI=SUM OF WEIGHT OF LIOH UTILIZED FROM LAST INCR(KG)
18 C      REMFACT=LIOH REMOVAL FACTOR (LB LIOH/LB CONTAMINANT)
19 C      OUTPUTS:
20 C      EFF=REMOVAL EFFICIENCY (DEC)
21 C
22 C      IF CONT DOESNT REACT WITH LIOH OR BEDL<=0 OR BED DIA <=0 OR
23 C      DENLI <=0 THEN REM EFF =0
24 C      IF((REMFACT.LE.0).OR.(BEDL.LE.0).OR.(BEDDIA.LE.0).OR.(DENLI.LE.0))
25 C      + THEN
26 C      EFF=0
27 C      ELSE
28 C      BED WEIGHT (KG)
29 C      BEDWGT=BEDL*(BEDDIA)**2*.785*DENLI
30 C      TOTAL WEIGHT OF LIOH UTILIZED AT AVERAGE TIME IN INCREMENT (KG)
31 C      TWUTLI=SWUTLI+RWUTLI*(TN-TN1)/2
32 C      IF (TWUTLI/BEDWGT.LE.0.8) THEN
33 C      EFF=EMAX
34 C      ELSE
35 C      EFF=EMAX*SIN((BEDWGT-TWUTLI)*1.57/(BEDWGT*0.2))
36 C      ENDIF
37 C      IF (BEDL.LT.0.0191) THEN
38 C      EFF=EFF*BEDL/0.0191
39 C      ENDIF
40 C      ENDIF
41 C      IF(EFF.LT.0) EFF=0
42 C      IF(EFF.GT.EMAX) EFF=EMAX
43 C      RETURN
44 C      END
45 C      *****  END OF SUBROUTINE ALIOH *****

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C      *****
2 C      *          SUBROUTINE CAFILL                      *
3 C      *  SUBROUTINE TO FILL ADJUSTABLE SIZE CHAR ARRAY WITH BLANKS  *
4 C      *****
5      SUBROUTINE CAFILL(NN,NROW,NCOL)
6      INTEGER NROW,NCOL
7      CHARACTER NN(NROW,NCOL)*30
8
9 C      NN=ARRAY NAME-ARRAY HAS 30 CHARACTERS
10 C     NROW=NUMBER OF ROWS IN ARRAY (INTEGER)
11 C     NCOL=NUMBER OF COLUMNS IN ARRAY(INTEGER)
12
13         DO 110 I=1,NROW
14         DO 100 J=1,NCOL
15             NN(I,J)=' '
16     100    CONTINUE
17     110    CONTINUE
18         RETURN
19 C     *****  END OF SUBROUTINE CAFILL  *****
20         END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C *****
2 C * SUBROUTINE CALCM *
3 C * SUBROUTINE TO CALCULATE SUM OF MASSES REMOVED BY ALL DEVICES *
4 C * USES CAV CABIN, REM EFF DD(J,20), & M.GEN DD(J,19) TO CALC *
5 C * DEVICE CIN & COUT DD(J,17)& DD(J,18),M.REM CABIN+DEV DD(J,21)*
6 C *****
7
8 SUBROUTINE CALCM(DD,NROW,NCOL,CAV,SMGEN,SMREM,TN,LIN2)
9 INTEGER NROW,NCOL,LIN2
10 REAL DD(NROW,NCOL)
11 C
12 C SUBROUTINES REQUIRED:
13 C NONE
14
15 C INPUTS:
16 C CAV=CABIN CONT AVERAGE CONCENTRATION (MG/CU M)
17 C DD,NROW,NCOL=NAME AND SIZE OF MAT DD
18 C TN=INCREMENT FINAL TIME (HRS)
19 C LIN2=NO. OF DEVICES IN MAT DD
20 C REM EFF AND DEVICE+CABIN M.GEN MUST BE LOADED INTO MAT DD
21 C BEFORE USING THIS SUBROUTINE
22 C OUTPUTS:
23 C M.REM (MG/HR) FOR ALL DEVICES + CABIN CALCULATED AT CAV,
24 C ARE STORED IN MAT DD COL 21
25 C SMREM=SUM OF MASS OF CONT REM IN DEVICES (MG/HR)
26 C SMGEN=SUM OF MASS GENERATED BY ALL DEVICES + CABIN (MG/HR) 27
27 C
28 C LOAD DEVICES 1 AND 2 WITH CIN AND COUT+M.REMOVED FOR DEV 2
29 C DEVICE 1=CABIN
30 C LOAD MAT DD WITH CAV CABIN
31 DD(1,17)=CAV
32 DD(1,18)=CAV
33 DD(1,21)=0
34 C DEVICE 2
35 DD(2,17)=CAV
36 DD(2,18)=CAV
37 DD(2,21)=DD(2,2)*DD(2,18)
38 C FOR DEV 3-15 CALC CIN COUT AND M.REMOVED BY DEVICE
39 C IF DEVICE FLOW RATE=0 THEN SET CIN,COUT,& M.GEN=0
40 DO 100 J=3,LIN2
41 IF (DD(J,2).EQ.0) THEN
42 DD(J,17)=0
43 DD(J,18)=0
44 DD(J,21)=0
45 GOTO 100
46 ENDIF
47 C IF UPSTREAM DEVICE=1 OR 2 THEN SET INLET=CABIN CONC+DEV M.GEN/Q

```

```

48      IF ((DD(J,4).EQ.1).OR.(DD(J,4).EQ.2)) THEN
49          DD(J,17)=DD(1,18)+DD(J,19)/DD(J,2)
50      ELSE
51 C      DETERMINE FLOWS,CIN AND COUT FOR DEVICES WITH RELATIVE ADDRESSES
52          IF (DD(J,4).EQ.0) THEN
53              QNO1=0
54              CNO1=0
55          ELSE
56              QNO1=DD(NINT(DD(J,4)),2)
57              CNO1=DD(NINT(DD(J,4)),18)
58          ENDIF
59          IF (DD(J,5).EQ.0) THEN
60              QNO2=0
61              CNO2=0
62          ELSE
63              QNO2=DD(NINT(DD(J,5)),2)
64              CNO2=DD(NINT(DD(J,5)),18)
65          ENDIF
66          IF (DD(J,6).EQ.0) THEN
67              QNO3=0
68              CNO3=0
69          ELSE
70              QNO3=DD(NINT(DD(J,6)),2)
71              CNO3=DD(NINT(DD(J,6)),18)
72          ENDIF
73 C
74 C      IF ALL UPSTREAM DEVICE FLOWS=0
75          IF (QNO1+QNO2+QNO3.EQ.0) THEN
76              DD(J,17)=0
77              DD(J,18)=0
78              DD(J,21)=0
79              OPEN(2,FILE='CON',IOSTAT=IOVAL)
80              WRITE(*,*) ' FLOW HALTED-UPSTREAM DEV TURNED OFF-
81 + INC END TIME;DEV=',TN,DD(J,1)
82              CLOSE (2)
83              GO TO 100
84          ELSE
85 C      CALCULATE CIN
86          DD(J,17)=(QNO1*CNO1+QNO2*CNO2+QNO3*CNO3)/
87 + (QNO1+QNO2+QNO3)+DD(J,19)/DD(J,2)
88          ENDIF
89 C      END OF DETERMINE FLOWS,CIN,COUT OF DEV WITH REL ADDR.
90      ENDIF
91 C      CALCULATE COUT
92          DD(J,18)=DD(J,17)*(1-DD(J,20))
93 C      CALCULATE SUM OF MASS REMOVED (CIN*Q*REM EFF)
94          DD(J,21)=DD(J,17)*DD(J,2)*DD(J,20)
95      100 CONTINUE
96 C      END OF LOADING OF MAT DD WITH DATA AND CALCULATING CIN COUT,M.REM
97 C      SUM TOTAL MASS OF CONT REMOVED BY ALL DEVICES (2-15) (MG/HR)

```

```

98          SMREM=0
99          DO 101 J=2,LIN2
100         SMREM=SMREM+DD(J,21)
101 101 CONTINUE
102 C      SUM MASS OF CONT GENERATED IN ALL DEVICES+CABIN (1-15) (MG/HR)
103         SMGEN=0
104         DO 102 J=1,LIN2
105         SMGEN=SMGEN+DD(J,19)
106 102 CONTINUE
107 C      CALC M.REM CABIN AND PUT IN DD(1,21)
108         DD(1,21)=SMGEN-SMREM
109         RETURN
110 C      ***** END OF SUBROUTINE CALCM *****
111         END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C      *****
2 C      *
3 C      *          SUBROUTINE CATBNR          *
4 C      *          CALCULATES EFFICIENCY OF CAT BURNER          *
5 C      *****
6
7 C      SUBROUTINE CATBNR(EMAX,OXID,EFF)
8 C
9 C      INPUTS:
10 C      EMAX=MAXIMUM BED EFFICIENCY (DEC)
11 C      OXID=DEGREE OF OXIDIZATION OF CHEMICAL (1=FULLY, 0=NONE)
12 C      OUTPUTS:
13 C      EFF=REMOVAL EFF (DEC)
14
15 C      IF(OXID.LT.0) OXID=0
16 C      IF(OXID.GT.1) OXID=1
17 C      EFF=EMAX*OXID
18 C      IF(EFF.LE.0) EFF=0
19 C      IF(EFF.GT.EMAX) EFF=EMAX
20 C      RETURN
21 C      END
22 C      ***** END OF SUBROUTINE CATBNR *****

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C *****
2 C * SUBROUTINE CNRSUB *
3 C * FOR 1 CONT AT A TIME AT CAV PRED, CALL EFF SUBROUTINES *
4 C * FOR DEVICES AND PUT EFFICIENCY IN MAT DD COL 20 *
5 C *****
6
7 C NOTE:DEVICE NUMBERS IN THIS SUBROUTINE REFER TO DEVICE TYPES,
8 C NOT THEIR RELATIVE POSITION IN MAT DD
9
10 C SUBROUTINE CNRSUB(I,TN,TN1,CAVPRD,DD,NROW,NCOL,CC,NROW1,NCOL1,
11 C SUBROUTINE CNRSUB(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
12 C +CDI,NROW2,NCOL2,LIN2)
13 C INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2
14 C REAL DD(NROW,NCOL)
15 C REAL CC(NROW1,NCOL1)
16 C REAL CDI(NROW2,NCOL2)
17 C
18 C SUBROUTINES REQUIRED:
19 C ACHBD-REMOVAL EFF OF AXIAL CHARCOAL BED
20 C RCHBD-REMOVAL EFF OF RADIAL CHARCOAL BED
21 C ALIOH-REMOVAL EFF OF AXIAL LIOH BED
22 C COOXID-REMOVAL EFF OF CO OXIDIZER
23 C CATBNR-REMOVAL EFF OF CAT BURNER
24 C CONDHX-REMOVAL EFF OF CONDENSING HX
25 C
26 C INPUTS:
27 C FROM PCSET PREDCT,AND CONVRG
28 C I=CONT NO.
29 C TN,TN1=CONT INCREMENT FINAL,INIT TIME (HRS)
30 C CAVPRD=CABIN AVERAGE CONT CONC (MG/CU M)
31 C DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
32 C CC,NROW1,NCOL1=NAME & DIM OF MAT CC
33 C CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
34 C LIN2=NUMBER OF DEVICES IN MAT DD
35 C FROM EFFICIENCY SUBROUTINES
36 C EFF=REMOVAL EFF (DEC)
37 C OUTPUTS:
38 C TO EFFICIENCY SUBROUTINES
39 C DEVICE AND CONTAMINANT INFORMATION AS REQUIRED
40 C DD(J,22)=DEVICE INLET CONCENTRATION
41 C TO PCSET, PREDCT, & CONVRG
42 C PUTS REMOVAL EFF FOR EACH DEVICE IN MAT DD COL 20
43 C
44 C EFFICIENCY FOR DEVICES 1 AND 2
45 C DD(1,20)=0
46 C DD(2,20)=DD(2,8)
47

```



```

48 C  SET COUNTER FOR READING CHAR SMR IN MAT CC FOR DEVICE TYPE 3
49      K=9
50 C
51 C  BEGIN LOOP FOR DEVICES 3 TO 15 OF MAT DD
52      DO 100 J=3,LIN2
53 C      INDEX COUNTER
54      K=K+3
55 C      SET REM EFF=0 AND GO TO END OF J LOOP IF DEVICE FLOW = 0
56      IF(DD(J,2).LE.1E-10) THEN
57          DD(J,20)=0
58          GOTO 80
59      ENDIF
60 C  DECISIONS FOR VARIOUS DEVICES
61      IF (NINT(DD(J,3)).EQ.3) THEN
62 C          GO TO SUBROUTINE FOR DEVICE TYPE 3-AXIAL CHARCOAL BED
63          CALL ACHBD(TN,TN1,DD(J,22),DD(1,10),DD(1,13),DD(J,2),DD(J,8),
64      + DD(J,9),DD(J,10),DD(J,12),DD(J,13),CDI(I,2),CDI(I,3),
65      + CDI(I,4),CDI(I,5),CDI(I,6),CC(I,K),EFF)
66 C          STORE EFF IN MAT DD
67          DD(J,20)=EFF
68      ELSEIF (NINT(DD(J,3)).EQ.4) THEN
69 C          GO TO SUBROUTINE FOR DEVICE TYPE 4-RADIAL CHARCOAL BED
70          CALL RCHBD(TN,TN1,DD(J,22),DD(1,10),DD(1,13),DD(J,2),DD(J,8),
71      + DD(J,9),DD(J,10),DD(J,11),DD(J,12),DD(J,13),CDI(I,2),CDI(I,3),
72      + CDI(I,4),CDI(I,5),CDI(I,6),CC(I,K),EFF)
73 C          STORE EFF IN MAT DD
74          DD(J,20)=EFF
75      ELSEIF (NINT(DD(J,3)).EQ.5) THEN
76 C          GO TO SUBROUTINE FOR DEVICE TYPE 5-LIOH BED
77          CALL ALIOH(TN,TN1,DD(J,8),DD(J,9),DD(J,10),DD(J,12),DD(J,15),
78      + DD(J,16),CDI(I,7),EFF)
79 C          STORE EFF IN MAT DD
80          DD(J,20)=EFF
81      ELSEIF (NINT(DD(J,3)).EQ.6) THEN
82 C          GO TO SUBROUTINE FOR DEVICE TYPE 6-CO OXIDIZER
83          CALL COOXID(DD(J,2),DD(J,8),DD(J,9),DD(J,10),CDI(I,4),EFF)
84 C          STORE EFF IN MAT DD
85          DD(J,20)=EFF
86      ELSEIF (NINT(DD(J,3)).EQ.7) THEN
87 C          GO TO SUBROUTINE FOR DEVICE TYPE 7-CAT BURNER
88          CALL CATBNR(DD(J,8),CDI(I,23),EFF)
89 C          STORE EFF IN MAT DD
90          DD(J,20)=EFF
91      ELSEIF (NINT(DD(J,3)).EQ.8) THEN
92 C          GO TO SUBROUTINE FOR DEVICE TYPE 8-CONDENSING HX
93          CALL CONDHX(DD(J,2),DD(J,8),DD(J,9),CDI(I,4),CDI(I,5),CDI(I,6),
94      + DD(J,22),EFF)
95 C          STORE EFF IN MAT DD
96          DD(J,20)=EFF
97      ELSEIF (NINT(DD(J,3)).EQ.9) THEN

```

```

98 C      SUBROUTINE FOR DEVICE TYPE 9-DUMMY
99        EFF=0
100 C     STORE EFF IN MAT DD
101        DD(J,20)=EFF
102        ELSE
103 C     DEVICES >9 OR <1 (OR ANY DEVICE NOT IN ABOVE CASES)
104        DD(J,20)=0
105        ENDIF
106 080 CONTINUE
107 100 CONTINUE
108        RETURN
109 C     ***** END OF SUBROUTINE CNRSUB *****
110        END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C      *****
2 C      *
3 C      *          SUBROUTINE CONDX
4 C      *          CALCULATES EFFICIENCY OF CONDENSING HX
5 C      *          *****
6          SUBROUTINE CONDX(BEDQ,EMAX,MLIQ,MW,VCONC,SOL,CAVPRD,EFF)
7          REAL MW,MLIQ,H,PA,XA,NOUT
8 C
9 C      INPUTS:
10 C      BEDQ=BED FLOW RATE (M3/HR)
11 C      EMAX=MAXIMUM POSSIBLE REMOVAL EFFICIENCY (DEC)
12 C      MLIQ=WATER FLOW RATE IN HX DUE TO CONDENSING (KG/HR)
13 C      MW=MOLECULAR WEIGHT OF CONTAMINANT
14 C      VCONT=VAPOR CONCENTRATION OF CONTAMINANT (MG/M3)
15 C      SOL=HENRY'S LAW COEFFICIENT (ATM/MOL FRACTION)
16 C      OUTPUTS:
17 C      EFF=REMOVAL EFF (DEC)
18 C
19          CAIN=CAVPRD
20          IF (CAIN.LE.1E-10) THEN
21              CAIN=0.1E-10
22          ENDIF
23          IF (SOL.LE.1E-10) THEN
24              EFF=0
25          ELSE
26 C      IF CONTAMINANT IS AMMONIA - USES EXPERIMENTAL DATA FROM JSC-08797
27 C      FOR AMMONIA REMOVAL AS A FUNCTION OF CO2 CONCENTRATION (8/23/76)
28          IF (MW.EQ.17.0) THEN
29              CAOUT=((CAIN*BEDQ)-(MLIQ*189.5847418*CAIN**0.534915256))/BEDQ
30              EFF=((CAIN-CAOUT)/CAIN)*EMAX
31          ELSE
32 C      CONTAMINANT IS NOT AMMONIA
33 C      CALCULATE CONTAMINANT PARTIAL PRESSURE AND WATER MOLE FRACTION
34              PA=CAIN*1.0E-9*82.06*278/MW
35              XA=(PA/1)/((MLIQ*(1000/18))/(BEDQ*(1000/22.4))+SOL/1)
36              NOUT=MLIQ*XA*1000/18
37              CAOUT=((CAIN*BEDQ)-(NOUT*MW*1000))/BEDQ
38              EFF=((CAIN-CAOUT)/CAIN)*EMAX
39          ENDIF
40          ENDIF
41          IF(EFF.LE.0) EFF=0
42          IF(EFF.GT.EMAX) EFF=EMAX
43          RETURN
44          END
45 C      ***** END OF SUBROUTINE CONDX *****

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS IN COMPILATION : 0

```

```

1 C *****
2 C * SUBROUTINE CONVRG *
3 C * MAIN CONVERGENCE LOOP SUBROUTINE *
4 C * USING CAV PRED & BASED ON SUM MASS REM OF LAST INCR, CALC *
5 C * NEW REM EFF, CAV CALC, CEQLIB, CFINAL, & M. REMOVED *
6 C * PUT THEM IN MAT DD - WORKS FOR ONE CONT AT A TIME *
7 C *****
8
9 SUBROUTINE CONVRG(I,TN,TN1,CAVPRD,DD,NROW,NCOL,
10 +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,KK,LIN,LIN2,
11 +NN,PRTSW3,IMSGDN)
12 INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2,PRTSW3,KK
13 CHARACTER NN(NROW1)*30
14 REAL DD(NROW,NCOL)
15 REAL CC(NROW1,NCOL1)
16 REAL CDI(NROW2,NCOL2)
17 REAL X1,X2,Y1,Y2,SLOPE
18
19 C SUBROUTINES REQUIRED:
20 C PRAFIL-ZERO MAT DD COL 17-21
21 C CNRSUB-USING CAV PRED CALC REM EFF FOR ALL DEVICES-PUT IN DD COL 20
22 C MASBAL-CALC CAV CALC,CFINAL,CEQLIB,M.REM
23 C INPUTS:
24 C FROM MCALC
25 C I=CONTAMINANT NO.
26 C TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
27 C DD,NROW,NCOL=NAME & DIM OF MAT DD
28 C CC,NROW1,NCOL1=NAME & DIM OF MAT CC
29 C CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
30 C LIN=NO. OF CONTAMINANTS IN MAT CDI
31 C LIN2=NO. DEVICES IN MAT DD
32 C IMSGDN=DEVICE NO. FOR MESSAGE OUTPUT
33 C CAVPRD=PREDICTED CABIN CONC FOR INCREMENT (MG/CU M)
34 C FROM PRAFIL
35 C ZEROS IN MAT DD COL 17-21
36 C FROM CNRSUB
37 C CALL REM EFF SUBROUTINE & PUTS REM EFF FOR EACH DEV IN DD COL 20
38 C FROM MASBAL
39 C CAVCLC=CALC CABIN CONT CONC (MG/CU M)
40 C CFINAL=FINAL INCR CABIN CONT CONC (MG/CUM)
41 C CEQLIB=EQUILIBRIUM CABIN CONT CONC (MG/CU M)
42 C OUTPUTS:
43 C TO MCALC
44 C CAVCLC=CALC CABIN CONT CONC (MG/CU M)
45 C CFINAL=FINAL INCR CABIN CONT CONC (MG/CUM)
46 C KK=COUNTER FOR CONVERGENCE
47 C CEQLIB=EQUILIBRIUM CABIN CONT CONC (MG/CU M)

```

```

48 C      M.REM IS IN MAT DD COL 21
49 C      TO PRAFIL
50 C      NAME & SIZE OF MATRIX + FIRST AND LAST COL TO BE ZEROED
51 C      TO CNRSUB
52 C      I=CONT NO.
53 C      TN,TN1=CONT INCREMENT FINAL,INIT TIME (HRS)
54 C      DEVICE AVERAGE CONT CONC (MG/M3) = DD(J,22)
55 C      DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
56 C      CC,NROW1,NCOL1=NAME & DIM OF MAT CC
57 C      CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
58 C      LIN2=NUMBER OF DEVICES IN MAT DD
59 C      TO MASBAL
60 C      I
61 C      TN,TN1
62 C      CVOL=CABIN VOL (CU M)=DD(1,9)
63 C      CINIT=INCR INIT CABIN CONT CONC (MG/CU M)=CC(I,1)
64 C
65 C      CONVERGENCE ERROR (DEC)
66 C      CNVERR=DD(1,12)
67 C      IF (CNVERR.LT.1E-10) THEN
68 C          OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
69 C          WRITE(IMSGDN,*) 'CONV ERROR<1E-10:PROGRAM TERMINATED'
70 C          CLOSE(IMSGDN)
71 C          STOP
72 C      ENDIF
73 C      KK=1
74 C      DO 100 KK=1,20
75 C          ZERO MAT DD COL 17-21
76 C          CALL PRAFIL(DD,NROW,NCOL,17,21)
77 C          USING CAVPRD FIND REM EFF OF EACH DEV & PUT IN DD COL 20
78 C          CALL CNRSUB(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
79 C      +      CDI,NROW2,NCOL2,LIN2)
80 C          FIND CAVCLC FOR THESE REMOVAL EFFICIENCIES
81 C          CALL MASBAL(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
82 C      +      CAVCLC,CDI,NROW2,NCOL2,CFINAL,CEQLIB,LIN,LIN2)
83 C          IF PRTSW3=1 THEN PRINT NAME & NO + CONV VALUES
84 C          IF (PRTSW3.EQ.1) THEN
85 C              OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
86 C              WRITE(IMSGDN,*) 'PRINTOUT FOR CONVERGENCE VALUES IN CONVRG'
87 C              WRITE(IMSGDN,50) I,NN(I)
88 C      050          FORMAT(1X,'CONT NO.= ',I4,2X,A)
89 C              WRITE(IMSGDN,*) 'CAVPRD,CAVCLC= ',CAVPRD,CAVCLC
90 C              CLOSE(IMSGDN)
91 C          ENDIF
92 C
93 C          IF CAVCLC=CAVPRD THEN EXIT THE KK LOOP
94 C          IF(CAVCLC.EQ.CAVPRD) GOTO 101
95 C          IF CAVPRD<1E-10 THEN SKIP CONVERGENCE STEP
96 C          IF(CAVPRD.LT.1E-10) GOTO 80
97 C          IF CONVERGENCE IS REACHED EXIT THE KK LOOP

```

```

98             IF (ABS((CAVPRD-CAVCLC)/CAVPRD).LT.CNVERR) THEN
99                 GOTO 101
100            ENDIF
101 C          CONVERGENCE CALCULATION ROUTINE
102 C          USE THE BISECTION METHOD FOR THE ITERATION WHERE KK=1
103 080      IF (KK.EQ.1) THEN
104 C          INITIALIZE X2 AND Y2 FOR THE NEXT ITERATION
105             X2=CAVPRD
106             Y2=CAVCLC-CAVPRD
107             CAVPRD=(CAVPRD+CAVCLC)/2
108         ELSE
109 C          USE THE NEWTON-RAPHSON METHOD FOR ITERATIONS WHERE KK>1
110             X1=X2
111             Y1=Y2
112             X2=CAVPRD
113             Y2=CAVCLC-CAVPRD
114             SLOPE=(Y2-Y1)/(X2-X1)
115             CAVPRD=X2-0.95*Y2/SLOPE
116         ENDIF
117 C
118 C          SET CAV IN PRED DD(I,22) = CAV IN CALC DD(I,17)
119             DO 90 J=1,LIN2
120                 DD(J,22)=DD(J,17)
121 090 CONTINUE
122 C
123 100 CONTINUE
124 C          END OF KK LOOP
125 101 CONTINUE
126 C
127
128             RETURN
129 C          ***** END OF SUBROUTINE CONVRG *****
130             END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C *****
2 C * SUBROUTINE COOXID *
3 C * CALCULATES EFFICIENCY OF CO OXIDIZER (Pt on charcoal) *
4 C *****
5 C
6 SUBROUTINE COOXID(BEDQ,EMAX,BEDL,BEDDIA,MW,EFF)
7 REAL MW
8 C
9 C INPUTS:
10 C BEDQ=BED FLOW RATE (CU M/HR)
11 C EMAX=MAXIMUM POSSIBLE REMOVAL EFFICIENCY (DEC)
12 C BEDL=BED LENGTH (M)
13 C BEDDIA=BED DIAMETER (M)
14 C MW=MOLECULAR WEIGHT OF CONTAMINANT
15 C OUTPUTS:
16 C EFF=REMOVAL EFF (DEC)
17 C
18 C WORKS ONLY FOR CO MW=28.01 OR H2=2.02; OTHERWISE REM EFF=0
19 C IF ((MW.EQ.28.01).OR.(MW.EQ.2.02)) THEN
20 C EFF=EMAX
21 C IF RESIDENCE TIME <0.2 SEC THEN REM EFF DROPS LINEARLY
22 C BREST = BED RESIDENCE TIME (SEC)
23 C BREST=(3.141592654/4)*BEDL*BEDDIA**2*3600/BEDQ
24 C IF (BREST.LT.0.2) THEN
25 C EFF=EMAX*BREST/0.2
26 C ENDIF
27 C ELSE
28 C REM EFF FOR OTHER THAN CO OR H2 = ZERO
29 C EFF=0
30 C ENDIF
31 C IF(EFF.LE.0) EFF=0
32 C IF(EFF.GT.EMAX) EFF=EMAX
33 C RETURN
34 C END
35 C ***** END OF SUBROUTINE COOXID *****

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS IN COMPILATION : 0

```

```

1 C      *****
2 C      *
3 C      *      SUBROUTINE  CRIN
4 C      *      SUBROUTINE TO READ STRING OF LENGTH 30 INTO MAT NN
5 C      *      AND READ REAL DATA INTO MAT XX(ROW,COL)
6 C      *      RETURNS NUMBER OF LINES OF DATA READ FROM FILE
7 C      *****
7 C      NOTE: INPUT STRING MUST HAVE SINGLE QUOTES AROUND IT
8 C      NOTE: INPUT NUMBERS MUST BE SEPARATED BY BLANKS
9      SUBROUTINE CRIN(NN,XX,NROW,NCOL,LIN)
10     INTEGER NROW,NCOL,IOVAL,LIN
11     CHARACTER NN(NROW)*30,FNAME*24
12     REAL XX(NROW,NCOL)
13     010 READ(*,'(A)') FNAME
14     OPEN(1,FILE=FNAME,STATUS='OLD',IOSTAT=IOVAL)
15     IF(IOVAL.NE.0) GOTO 900
16     LIN=0
17     DO 100 I=1,NROW
18     READ(1,*,IOSTAT=IOVAL,END=500,ERR=900 ) NN(I),(XX(I,J),J=1,NCOL)
19     LIN=LIN+1
20     100 CONTINUE
21     500 WRITE(*,'(A)') ' DONE WITH FILE INPUT'
22     WRITE (*,*) ' '
23     CLOSE (1)
24     GOTO 990
25     900 WRITE(*,*) 'IOERROR= ',IOVAL
26     CLOSE (1)
27     WRITE(*,*) 'WHAT IS THE INPUT FILE NAME? '
28     GOTO 10
29     990 RETURN
30 C      ***** END OF SUBROUTINE CRIN *****
31     END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```



```

1 C      *****
2 C      *          SUBROUTINE  CROUT                      *
3 C      *          SUBROUTINE TO WRITE DATA TO CONSOLE, OR PRINTER          *
4 C      *          WRITES STRG OF LENGTH 30 FROM MAT NN & REAL DATA FROM MAT  *
5 C      *          XX(ROW,COL) STARTING WITH COL FSTCOL, AND ENDING WITH LSTCOL *
6 C      *          AND FROM LINE FSTLIN TO LINE LSTLIN                      *
7 C      *****
8          SUBROUTINE CROUT(NN,XX,NROW,NCOL,FSTCOL,LSTCOL,LIN,FSTLIN,LSTLIN,
9          +IMSGDN,NINC,FNAME,IDEVNO,IOVAL)
10         INTEGER NROW,NCOL,IOVAL,FSTCOL,LSTCOL,LIN,FSTLIN,LSTLIN,NINC,IDEVNO,
11         + IOVAL
12         CHARACTER FNAME*24,DES*1
13         CHARACTER NN(NROW)*30
14         REAL XX(NROW,NCOL)
15         IF (FSTCOL.GT.NCOL) FSTCOL=NCOL
16         IF (LSTCOL.GT.NCOL) LSTCOL=NCOL
17         IF (FSTCOL.GT.LSTCOL) FSTCOL=LSTCOL
18         IF (FSTLIN.GT.LIN) FSTLIN=LIN
19         IF (LSTLIN.GT.LIN) LSTLIN=LIN
20         IF (FSTLIN.GT.LSTLIN) FSTLIN=LSTLIN
21
22 C 010 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
23 C      WRITE(IMSGDN,'(A)') ' WRITE TO LPT1 OR CON OR END
24 C      CLOSE (IMSGDN)
25 C      READ(*,'(A)') FNAME
26 C      QUIT IF FNAME=END
27 C      IF (FNAME.EQ.'END') GO TO 990
28 C      IF((FNAME.NE.'LPT1').AND.(FNAME.NE.'CON')) GOTO 10
29 C      OPEN(1,FILE=FNAME,IOSTAT=IOVAL)
30 C      IF(IOVAL.NE.0) GOTO 900
31 C      WRITE (IDEVNO,55,IOSTAT=IOVAL,ERR=900) NINC
32 C 055      FORMAT ('INCREMENT NO. = ',I7)
33 C      DO 110 I=FSTLIN,LSTLIN
34 C      WRITE(1,60,IOSTAT=IOVAL,ERR=900) I,NN(I)
35 C      WRITE(IDEVNO,60,IOSTAT=IOVAL,ERR=900) I,NN(I)
36 C 060      FORMAT(1X,'CONT NO.= ',I4,2X,A)
37 C      WRITE(1,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
38 C      WRITE(IDEVNO,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
39 C 070      FORMAT(1X,7G11.4)
40 C 110 CONTINUE
41 C      CLOSE (1)
42 C      GOTO 990
43 C 900 OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
44 C      WRITE(IMSGDN,*) 'IOERROR= ',IOVAL
45 C      CLOSE(IMSGDN)
46 C      CLOSE (1)
47 C      CLOSE (IDEVNO)

```

```
48 C      GOTO 10
49 990 RETURN
50 C      ***** END OF SUBROUTINE CROUT *****
51      END
```

```
NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```

1 C      *****
2 C      *          SUBROUTINE  CROUT2                      *
3 C      *          SUBROUTINE TO WRITE DATA TO CONSOLE, OR PRINTER      *
4 C      *          WRITES STRG OF LENGTH 30 FROM MAT NN & REAL DATA FROM MAT  *
5 C      *          XX(ROW,COL) STARTING WITH COL FSTCOL, AND ENDING WITH LSTCOL  *
6 C      *          AND FROM LINE FSTLIN TO LINE LSTLIN                      *
7 C      *****
8          SUBROUTINE CROUT2 (NN,XX,NROW,NCOL,FSTCOL,LSTCOL,LIN,FSTLIN,LSTLIN,
9          +IMSGDN)
10         INTEGER NROW,NCOL,IOVAL,FSTCOL,LSTCOL,LIN,FSTLIN,LSTLIN
11         CHARACTER FNAME*24,DES*1
12         CHARACTER NN(NROW)*30
13         REAL XX(NROW,NCOL)
14         IF (FSTCOL.GT.NCOL) FSTCOL=NCOL
15         IF (LSTCOL.GT.NCOL) LSTCOL=NCOL
16         IF (FSTCOL.GT.LSTCOL) FSTCOL=LSTCOL
17         IF (FSTLIN.GT.LIN) FSTLIN=LIN
18         IF (LSTLIN.GT.LIN) LSTLIN=LIN
19         IF (FSTLIN.GT.LSTLIN) FSTLIN=LSTLIN
20
21         010 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
22             WRITE(IMSGDN,'(A)') ' WRITE TO LPT1 OR CON OR END '
23             CLOSE (IMSGDN)
24             READ(*,'(A)') FNAME
25             QUIT IF FNAME=END
26             IF (FNAME.EQ.'END') GO TO 990
27             IF ((FNAME.NE.'LPT1').AND.(FNAME.NE.'CON')) GOTO 10
28             OPEN(1,FILE=FNAME,IOSTAT=IOVAL)
29             IF(IOVAL.NE.0) GOTO 900
30             DO 110 I=FSTLIN,LSTLIN
31                 WRITE(1,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
32             070   FORMAT(1X,7G11.4)
33             110 CONTINUE
34             CLOSE (1)
35             GOTO 990
36         900 OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
37             WRITE(IMSGDN,*) 'IOERROR= ',IOVAL
38             CLOSE(IMSGDN)
39             CLOSE (1)
40             GOTO 10
41         990 RETURN
42 C      ***** END OF SUBROUTINE CROUT *****
43         END

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0

NUMBER OF ERRORS IN PROGRAM UNIT: 0

NUMBER OF WARNINGS IN COMPILATION : 0

NUMBER OF ERRORS IN COMPILATION : 0

```

1 C      *****
2 C      *      SUBROUTINE DATOUT      *
3 C      * SUBROUTINE TO PRINT HEADINGS AND DATA TO PRINTER,CON,OR FILE *
4 C      *****
5 C      NOTES: (1) FILE MUST BE OPEN BEFORE CALLING THIS SUBROUTINE
6 C      (2) IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED
7
8      SUBROUTINE DATOUT(TN,TN1,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,
9      +CDI,NROW2,NCOL2,LIN2,NN,PRTSW6,PRTSW8,PRTSW9,
10     +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR,
11     +TVAL,FCPLOT,IDEVN1,IDEVN3,IDEVN2)
12
13     INTEGER PRTSW6,PRTSW8,PRTSW9,TVAL,NINC
14
15 C      SUBROUTINES REQUIRED:
16 C      PRCD=PRINT OUT OF CONTAMINANT DATA
17 C      PRREM1=PRINTOUT OF RATE OF CONTAMINANT REMOVAL (MG/HR)-SHEET1
18 C      PRREM2=PRINTOUT OF RATE OF CONTAMINANT REMOVAL (MG/HR)-SHEET2
19 C      PRMAS1=PRINTOUT OF SUM MASSES REMOVED BY DEVICES (MG)-SHEET1
20 C      PRMAS2=PRINTOUT OF SUM MASSES REMOVED BY DEVICES (MG)-SHEET2
21 C      PREF=PRINTOUT OF INCREMENT END REMOVAL EFFICIENCIES
22
23 C      INPUTS FROM MAIN PROGRAM:
24 C      TN=INCREMENT FINAL TIME (HRS)
25 C      TN1=INCREMENT INITIAL TIME (HRS)
26 C      LIN=NO. OF CONT IN MAT CC A D NN
27 C      DD,NROW,NCOL=NAME & SIZE OF MAT DD
28 C      CC,NROW1,NCOL1=NAME & SIZE F MAT
29 C      CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
30 C      LIN2=NO. DEVICES IN MAT DD
31 C      NN=NAME OF MAT NN
32 C      NINC=TIME INCREMENT NUMBER
33 C      =0 THEN PRINT HEADINGS & DATA FOR PRECALCULATION SET UP ROUTINE
34 C      =-1 THEN PRINT HEADINGS & DATA FOR FINAL ANSWERS
35 C      ELSE PRINT WITH PROPER INCREMENT NUMBER
36 C      IDEVNO=OUTPUT DEVICE NUMBER (SHOULD BE 6)
37 C      IMONTH..IHOUR=DATE AND TIME VARIABLES
38 C      IPGCTR=PAGE COUNTER FOR SEQUENTIAL PAGE NO.'S ON ALL PAGES
39
40 C      OUTPUT TO MAIN PROG:
41 C      IOVAL=STATUS OF IOERROR IN SUBROUTINES
42
43 *      TEST CASE *****
44 *      LIN=120
45
46 C      PRINT OUT CONCENTRATION DATA
47     CALL PRCD(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,

```

```

48      + IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR,
49      + PRTSW8,PRTSW9,FCPLOT,IDEVN1)
50 C    PRINT OUT NHB 8060.1 GROUP CONTRIBUTION VALUES (T-VALUES)
51      IF ((TVAL.EQ.1).OR.(TVAL.EQ.2)) THEN
52      CALL GROUP(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
53      + IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,
54      + IPGCTR,TVAL,IDEVN3,PRTSW8)
55      ENDIF
56      IF (PRTSW6.EQ.1) THEN
57      IF ((PRTSW8.EQ.1).OR.((PRTSW8.EQ.0).AND.(NINC.EQ.-1))) THEN
58 C    PRINTOUT OF RATE OF CONTAMINANT REMOVAL BY DEVICES-SHEET1
59      CALL PRREM1(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
60      + IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)
61 C    DON'T PRINT SHEET 2 UNLESS NUMBER OF DEVICES IN MAT DD > 8
62      IF (LIN2.GT.8) THEN
63 C    PRINTOUT OF RATE OF CONTAMINANT REMOVAL BY DEVICES-SHEET2
64      CALL PRREM2(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
65      + IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)
66      ENDIF
67 C    PRINTOUT OF SUM OF MASS REMOVED BY DEVICES-SHEET1
68      CALL PRMAS1(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
69      + IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)
70 C    DON'T PRINT SHEET 2 UNLESS NUMBER OF DEVICES IN MAT DD > 8
71      IF (LIN2.GT.8) THEN
72 C    PRINTOUT OF SUM OF MASS REMOVED BY DEVICES-SHEET2
73      CALL PRMAS2(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
74      + IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)
75      ENDIF
76      ENDIF
77 C    PRINTOUT OF INCREMENT END REMOVAL EFFICIENCIES
78      CALL PREFF(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
79      + IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,
80      + IPGCTR,PRTSW8,PRTSW9,IDEVN2)
81      ENDIF
82
83      RETURN
84      END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C  FILE GROUP
2 C      *****
3 C      *          SUBROUTINE GROUP          *
4 C      * PROGRAM TO PRINT THE GROUP TOXICITY LEVELS AND T LEVEL *
5 C      *****
6 C
7 C
8      SUBROUTINE GROUP(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
9      +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,
10     +IPGCTR,TVAL,IDEVNT,PRTSW8)
11 C
12 C
13     DIMENSION GL(16)
14     REAL CC(NROW1,NCOL1)
15     REAL CDI(NROW2,NCOL2)
16     REAL TLEVL
17     INTEGER TVAL,PRTSW8,NINC,IDEVNO,IDEVNT
18 C     DETERMINE THE SUMS FOR EACH GROUP LEVEL
19     DO 25 J=1,16
20     25  GL(J)=0.
21         DO 30 I=1,LIN
22             FRACT = CC(I,4)/CDI(I,9)
23             TSTR = CDI(I,8)
24             NHB = IFIX(TSTR)
25             GL(NHB) = GL(NHB) + FRACT
26     30  CONTINUE
27 C     CALCULATE THE TLEVEL OF THE ASSOCIATED GROUP LEVELS
28     TLEVL = GL(1)+GL(2)+GL(3)+GL(4)+GL(5)+GL(9)+GL(10)+GL(11)+
29     + GL(13)+GL(14)+GL(16)
30     IF ((PRTSW8.EQ.1).OR.((PRTSW8.EQ.0).AND.(NINC.EQ.-1))) THEN
31         WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900) ' '
32         WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900) ' '
33         WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900)
34     +' GROUP T-VALUES AS SPECIFIED IN NHB 8060.1B APPENDIX D'
35         WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900) ' '
36         WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900)
37     +' -01-   -02-   -03-   -04-   -05-   -06-   -07-   -08-   -09-
38     + -10-   -11-   -12-   -13-   -14-   -15-   -16-'
39         WRITE (IDEVNO,'(1X,16(F6.2,1X))',IOSTAT=IOVAL,ERR=900) GL(1),
40     + GL(2),GL(3),GL(4),GL(5),GL(6),GL(7),GL(8),GL(9),GL(10),
41     + GL(11),GL(12),GL(13),GL(14),GL(15),GL(16)
42         WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900) ' OVERALL T-VALUE'
43         OALLT = GL(1)+GL(2)+GL(3)+GL(4)+GL(5)+GL(9)+GL(10)+GL(11)+
44     + GL(13)+GL(14)+GL(16)
45         WRITE (IDEVNO,'(T4,F7.2)',IOSTAT=IOVAL,ERR=900) OALLT
46         ENDIF
47         IF (NINC.NE.-1) THEN

```

```

48 C      ***** WRITE T-VALUE DATA TO A PLOT FILE *****
49          IF (TVAL.EQ.2) THEN
50              WRITE (IDEVNT,50,IOSTAT=IOVAL,ERR=900) TN1,TN,TLEVL
51      50          FORMAT (T2,2(F8.2,1X),F7.2)
52              ENDIF
53          ENDIF
54          GOTO 999
55 C      ***** END OF SUBROUTINE *****
56      900 WRITE(*,*)'IO ERROR IN GROUP= ',IOVAL
57      999 RETURN
58          END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C  FILE:HEADGS.FOR
2 C  *****
3 C  *          SUBROUTINE HDG1                      *
4 C  * PROGRAM TO PRINT HEADING-DATE, TIME, FILE NAME, & PAGE NO. *
5 C  *****
6
7 C  NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
8
9  SUBROUTINE HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,
10 +FNAME,IPGNO,IDEVNO)
11 C  IMONTH...ISECOND=TIME AND DATE NAMES
12 C  FNAME=FILE NAME
13 C  IPGNO=PAGE NUMBER
14 C  IDEVNO=DEVICE NUMBER FOR OUTPUT
15
16  CHARACTER FNAME*24
17
18
19 C  WRITE HEADING
20  IF (IPGNO .EQ. 1) THEN
21  WRITE (IDEVNO,5,IOSTAT=IOVAL,ERR=900)
22  005  FORMAT (2X,'PROGRAM VERSION 8.0 Alpha',5X,'04-10-92',/)
23  ENDIF
24  WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) IMONTH,IDAY,IYEAR,
25  +IHOUR,IMINUTE,FNAME,IPGNO
26  010  FORMAT(2X,I2,'/',I2,'/',I4,5X,I2,':',I2,5X,A,2X,'PAGE ',I4)
27
28  GO TO 999
29  900 WRITE(*,*)'IO ERROR IN HDG1= ',IOVAL
30  999 RETURN
31  END

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0

NUMBER OF ERRORS IN PROGRAM UNIT: 0


```

32
33 C *****
34 C *          SUBROUTINE DATTM                      *
35 C * SUBROUTINE TO READ SYSTEM DATE AND TIME FOR IBM PC OR AT *
36 C *****
37
38 SUBROUTINE DATTM(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,ISECOND)
39
40 C REQUIRED FOR IBM PROF FORTRAN
41 INTEGER*2 IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,ISECOND,IHUNSEC
42
43 CALL GETDAT(IYEAR,IMONTH,IDAY)
44 CALL GETTIM(IHOUR,IMINUTE,ISECOND,IHUNSEC)
45
46 RETURN
47 END

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS IN PROGRAM UNIT: 0

```

48
49 C *****
50 C *          SUBROUTINE HDG2                      *
51 C * PROGRAM TO PRINT HEADING-TIME INCR+INCR INIT AND FINAL TIME *
52 C * PRINTS TIME INCREMENT NUMBER                      *
53 C *****
54
55 C NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
56
57 SUBROUTINE HDG2(INCRNO,TN1,TN,IDEVNO)
58
59 C INPUTS:
60 C INCRNO=TIME INCREMENT NUMBER
61 C TN1=INCREMENT INITIAL TIME (HRS)
62 C TN=INCREMENT FINAL TIME (HRS)
63 C IDEVNO=DEVICE NUMBER FOR OUTPUT
64
65 C WRITE HEADING
66 WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) INCRNO,TN1,TN
67 010 FORMAT(1X,'TIME INCR ',I5,2X,'INITIAL TIME (HRS)= ',F8.2,2X,
68 + 'FINAL TIME (HRS)= ',F8.2)
69
70 GO TO 999
71 900 WRITE(*,*)'IO ERROR IN HDG2= ',IOVAL
72 999 RETURN
73 END

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS IN PROGRAM UNIT: 0

```

74
75 C *****
76 C *          SUBROUTINE HDG3                      *
77 C * PROGRAM TO PRINT HEADING-TIME INCR+INCR INIT AND FINAL TIME *
78 C * PRINTS PCALC OR FINAL INSTEAD OF TIME INCREMENT NUMBER      *
79 C *****
80
81 C NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
82
83 SUBROUTINE HDG3 (IFLAG,TN1,TN,IDEVNO)
84
85 C INPUTS:
86 C IFLAG=FLAG FOR TIME INCREMENT (1=PCALC, 2=FINAL)
87 C TN1=INCREMENT INITIAL TIME (HRS)
88 C TN=INCREMENT FINAL TIME (HRS)
89 C IDEVNO=DEVICE NUMBER FOR OUTPUT
90
91 CHARACTER INAME*5
92
93 IF (IFLAG.EQ.1) THEN
94   INAME='PCALC'
95 ELSEIF (IFLAG.EQ.2) THEN
96   INAME='FINAL'
97 ELSE
98   INAME='ERROR'
99 ENDF
100
101 C WRITE HEADING
102 WRITE (IDEVNO,10,IOSTAT=IOVAL,ERR=900) INAME,TN1,TN
103 010 FORMAT(1X,'TIME INCR ',A,2X,'INITIAL TIME (HRS)= ',F8.2,2X,
104 + 'FINAL TIME (HRS)= ',F8.2)
105
106 GO TO 999
107 900 WRITE(*,*) 'IO ERROR IN HDG3= ',IOVAL
108 999 RETURN
109 END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

```

```

110
111 C *****
112 C *          SUBROUTINE HDG4                      *
113 C * PROGRAM TO PRINT HEADING-CONT NO., NAME, FINAL CABIN CONC *
114 C * MAC, EXCEEDS MAC                             *
115 C *****
116
117 C NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
118
119 SUBROUTINE HDG4(IDEVNO)
120
121 C INPUTS:
122 C IDEVNO=DEVICE NUMBER FOR OUTPUT
123
124 C WRITE HEADING
125 WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)
126 010 FORMAT(1X,'CONT',14X,'NAME',14X,'FINAL CABIN',5X,'MAC',5X,
127 + 'EXCEEDS')
128 WRITE(IDEVNO,20,IOSTAT=IOERR,ERR=900)
129 020 FORMAT(1X,' NO.',32X,'CONC (MG/M3)',12X,' MAC ')
130
131 GO TO 999
132 900 WRITE(*,*)'IO ERROR IN HDG4= ',IOVAL
133 999 RETURN
134 END

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS IN PROGRAM UNIT: 0

```

135
136 C *****
137 C *          SUBROUTINE HDG5                      *
138 C * PROGRAM TO PRINT HEADING-TOTAL CONT REMOVED BY EACH DEV (MG) *
139 C * PRINTS SHEET 1-NO,NAME,CABIN,LEAK,&DEV3..DEV8 *
140 C *****
141
142 C NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
143
144 SUBROUTINE HDG5(IDEVNO)
145
146 C INPUTS:
147 C IDEVNO=DEVICE NUMBER FOR OUTPUT
148
149 C WRITE HEADING
150 WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)
151 010 FORMAT(1X,24X,'TOTAL CONTAMINANT MASS REMOVED BY EACH DEVICE (MG)
152 + SHEET 1')
153 WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
154 020 FORMAT(1X,' NO.',14X,'NAME',16X,'CABIN',9X,'LEAK',8X,'DEV3',
155 +8X,'DEV4',8X,'DEV5',8X,'DEV6',8X,'DEV7',8X,'DEV8')

```

```

156
157      GO TO 999
158      900 WRITE(*,*) 'IO ERROR IN HDG5= ',IOVAL
159      999 RETURN
160      END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

```

```

161
162 C      *****
163 C      *          SUBROUTINE HDG6                      *
164 C      * PROGRAM TO PRINT HEADING-TOTAL CONT REMOVED BY EACH DEV (MG) *
165 C      * PRINTS SHEET 2-NO,NAME,& DEV9..DEV15          *
166 C      *****
167
168 C      NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
169
170      SUBROUTINE HDG6(IDEVNO)
171
172 C      INPUTS:
173 C      IDEVNO=DEVICE NUMBER FOR OUTPUT
174
175 C      WRITE HEADING
176      WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)
177      010 FORMAT(1X,24X,'TOTAL CONTAMINANT MASS REMOVED BY EACH DEVICE (MG)
178      +          SHEET 2')
179      WRITE(IDEVNO,20,IOSTAT=IOERR,ERR=900)
180      020 FORMAT(1X,' NO.',14X,'NAME',16X,' DEV9',8X,'DEV10',7X,'DEV11',
181      +7X,'DEV12',7X,'DEV13',7X,'DEV14',7X,'DEV15')
182
183      GO TO 999
184      900 WRITE(*,*) 'IO ERROR IN HDG6= ',IOVAL
185      999 RETURN
186      END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

```

```

187
188 C      *****
189 C      *          SUBROUTINE HDG7                      *
190 C      * PROGRAM TO PRINT HEADING-DEVICE REM EFF AT END OF TIME INCR *
191 C      * PRINTS NO.,NAME,#2..#12                      *
192 C      *****
193
194 C      NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
195
196      SUBROUTINE HDG7(IDEVNO)
197

```

```

198 C   INPUTS:
199 C       IDEVNO=DEVICE NUMBER FOR OUTPUT
200
201 C       WRITE HEADING
202         WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)
203 010 FORMAT(1X,24X,'DEVICE REMOVAL EFFICIENCY AT END OF TIME INCREMENT
204       + (DEC)')
205         WRITE(IDEVNO,20,IOSTAT=IOERR,ERR=900)
206 020 FORMAT(1X,' NO.',14X,'NAME',15X,'#2',4X,'#3',
207       +4X,'#4',4X,'#5',4X,'#6',4X,'#7',4X,'#8',4X,'#9',3X,'#10',
208       +3X,'#11',3X,'#12',3X,'#13',3X,'#14',3X,'#15')
209
210       GO TO 999
211 900 WRITE(*,*)'IO ERROR IN HDG7= ',IOVAL
212 999 RETURN
213       END

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS IN PROGRAM UNIT: 0

```

214
215 C   *****
216 C   *           SUBROUTINE HDG8                               *
217 C   * PROGRAM TO PRINT HEADING-RATE OF CONT REMOVAL-EACH DEV .(MG) *
218 C   * PRINTS SHEET 1-NO,NAME,CABIN,LEAK,&DEV3..DEV8           *
219 C   *****
220
221 C   NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
222
223       SUBROUTINE HDG8(IDEVNO)
224
225 C   INPUTS:
226 C       IDEVNO=DEVICE NUMBER FOR OUTPUT
227
228 C       WRITE HEADING
229         WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)
230 010 FORMAT(1X,24X,'RATE OF CONTAMINANT REMOVAL-EACH DEVICE (MG/HR)
231       + SHEET 1')
232         WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
233 020 FORMAT(1X,' NO.',14X,'NAME',16X,'CABIN',9X,'LEAK',8X,'DEV3',
234       +8X,'DEV4',8X,'DEV5',8X,'DEV6',8X,'DEV7',8X,'DEV8')
235
236       GO TO 999
237 900 WRITE(*,*)'IO ERROR IN HDG8= ',IOVAL
238 999 RETURN
239       END

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS IN PROGRAM UNIT: 0

```

240
241 C      *****
242 C      *          SUBROUTINE HDG9          *
243 C      * PROGRAM TO PRINT HEADING-RATE OF CONT REMOVAL-EACH DEV(MG/HR) *
244 C      * PRINTS SHEET 2-NO,NAME,& DEV9..DEV15          *
245 C      *****
246
247 C      NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
248
249      SUBROUTINE HDG9(IDEVNO)
250
251 C      INPUTS:
252 C      IDEVNO=DEVICE NUMBER FOR OUTPUT
253
254 C      WRITE HEADING
255      WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)
256      010 FORMAT(1X,24X,'RATE OF CONTAMINANT REMOVAL-EACH DEVICE (MG/HR)
257      +          SHEET 2')
258      WRITE(IDEVNO,20,IOSTAT=IOERR,ERR=900)
259      020 FORMAT(1X,' NO.',14X,'NAME',16X,' DEV9',8X,'DEV10',7X,'DEV11',
260      +7X,'DEV12',7X,'DEV13',7X,'DEV14',7X,'DEV15')
261
262      GO TO 999
263      900 WRITE(*,*)'IO ERROR IN HDG9= ',IOVAL
264      999 RETURN
265      END
266

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C      *****
2 C      *          SUBROUTINE LDIGEN                      *
3 C      *  SUBROUTINE TO LOAD INTERNAL GENERATION FROM MAT CDI COL 1 & *
4 C      *  COL 10-22 INTO MAT DD COL 19                    *
5 C      *****
6      SUBROUTINE LDIGEN(I,DD,NROW,NCOL,CDI,NROW2,NCOL2,LIN2)
7      INTEGER NROW,NCOL,NROW2,NCOL2,LIN2
8      REAL DD(NROW,NCOL)
9      REAL CDI(NROW2,NCOL2)
10
11 C  INPUTS:
12 C      I=CONTAMINANT NUMBER
13 C      DD,NROW,NCOL=NAME AND DIMENSIONS OF MAT DD
14 C      CDI,NROW2,NCOL2=NAME AND DIMENSIONS OF MAT CDI
15 C      LIN2=NUMBER OF DEVICES IN MAT DD
16 C  OUTPUT
17 C      LOADS INTERNAL GENERATION FROM MAT CDI INTO MAT DD COL 19
18 C
19      DD(1,19)=CDI(I,1)
20      DD(2,19)=0
21      DO 10 J=3,LIN2
22          DD(J,19)=CDI(I,J+7)
23 010 CONTINUE
24      RETURN
25 C      ***** END OF SUBROUTINE LDIGEN *****
26      END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C      *****
2 C      *              SUBROUTINE LODEFF              *
3 C      * SUBROUTINE TO LOAD LAST INCR EFF FROM MAT CC INTO MAT DD COL 20*
4 C      *      USES ADJUSTABLE SIZE ARRAYS          *
5 C      *****
6      SUBROUTINE LODEFF(I,DD,NROW,NCOL,CC,NROW1,NCOL1,LIN2)
7      INTEGER NROW,NCOL,NROW1,NCOL1
8      REAL DD(NROW,NCOL)
9      REAL CC(NROW1,NCOL1)
10
11 C      INPUTS:
12 C      I=CONTAMINANT LINE NUMBER IN MAT CC
13 C      DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
14 C      CC,NROW1,NCOL1=NAME & DIMENSIONS OF MAT CC
15 C      LIN2=NO. OF DEVICES IN MAT DD
16
17      DD(1,20)=0
18      K=7
19      DO 100 J=2,LIN2
20          DD(J,20)=CC(I,K)
21          K=K+3
22 100 CONTINUE
23      RETURN
24 C      ***** END OF SUBROUTINE LODEFF *****
25      END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```



```

1 C *****
2 C *          MASS BALANCE SUBROUTINE-MASBAL          *
3 C * FOR 1 CONT AT A TIME AT A GIVEN DEVICE EFFICIENCY CALCULATES *
4 C * CAV,CFINAL,CEQ,M.REMOVED (ALL DEV+CABIN)-DATA PUT IN MAT DD *
5 C *****
6 C NOTE: BEFORE RUNNING THIS SUBROUTINE MUST ZERO MAT DD COL 17-21
7 C (DONE BY PRAFIL) & LOAD REM EFF FOR EACH DEVICE INTO
8 C MAT DD COL 20 (DONE BY LODEFF OR CNRSUB)
9
10 SUBROUTINE MASBAL(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
11 +CAVCLC,CDI,NROW2,NCOL2,CFINAL,CEQLIB,LIN,LIN2)
12 INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2
13 REAL DD(NROW,NCOL)
14 REAL CC(NROW1,NCOL1)
15 REAL CDI(NROW2,NCOL2)
16 C
17 C SUBROUTINES REQUIRED:
18 C CALCM-CALCULATE CIN,COUT,M.REM,SUM MASS REM-IN MAT DD
19 C LDIGEN-LOAD INTERNAL M.GEN FOR DEVICE + CABIN FROM CDI INTO DD COL 19
20 C PCAVCF-USING CEQ & CINIT, CALC CFINAL & CAVERAGE
21
22 C INPUTS:
23 C FROM PCSET, PREDCT, AND CONVRG
24 C I=CONTAMINANT NO.
25 C TN=INCREMENT END TIME(HRS); TNI=INCR BEGINNING TIME HRS
26 C DD,NROW,NCOL=NAME AND SIZE OF MAT DD
27 C CC,NROW1,NCOL1=NAME AND SIZE OF MAT CC
28 C CDI,NROW2,NCOL2=NAME AND SIZE OF MAT CDI
29 C LIN=NO. OF CONT IN MAT CDI
30 C LIN2=NO. OF DEVICES IN MAT DD
31 C FROM CALCM
32 C SMREM=SUM OF MASS REM FOR ALL DEVICES (MG/HR)-TOTAL OF DD COL 21
33 C SMGEN=SUM OF MASS GEN IN ALL DEVICES INCL CABIN(MG/HR)-DD COL 19
34 C FROM LDIGEN
35 C IT LOADS CABIN M.GEN (MG/HR) FROM MAT CDI INTO DD(1,19)
36 C IT LOADS M.GEN DEVICES FROM MAT CDI COL 2-15,19 INTO DD COL 19
37 C FROM PCAVCF
38 C CAVCLC=CALC INCR CABIN CONT CONC (MG/CU M)
39 C CFINAL=FINAL INCR CABIN CONT CONC (MG/CU M)
40 C OUTPUTS:
41 C TO PCSET, PREDCT, AND CONVRG
42 C CAVCLC=CALCULATED CABIN AVERAGE CONC(MG/CU M)
43 C CEQLIB=CABIN EQUILIBRIUM CONCENTRATION (MG/CU M)
44 C CFINAL=INCREMENT FINAL CABIN CONCENTRATION (MG/CU M)
45 C PUTS M.REM FOR CABIN + DEVICES IN MAT DD COL 21
46 C TO CALCM
47 C CAV=CABIN CONT CONC (MG/CU M)

```

```

48 C      DD(1,19)=50 (CABIN M.GEN)
49 C      OTHER DEVICES DD(2-15,19) MUST =0 AT THIS POINT (SEE PRAFIL)
50 C      TO LDIGEN
51 C      I=CONTAMINANT NUMBER
52 C      DD,NROW,NCOL=NAME AND DIMENSIONS OF MAT DD
53 C      CDI,NROW2,NCOL2=NAME AND DIMENSIONS OF MAT CDI
54 C      LIN2=NUMBER OF DEVICES IN MAT DD
55 C      TO PCAVCF
56 C      TN,TN1
57 C      CINIT=INITIAL INCR CABIN CONT CONC (MG/CU M)=CC(I,1)
58 C      CEQLIB=CABIN EQUILIB CONC (MG/CU M)
59 C      SQEFFN=SUM OF Q*REM EFF NET FOR ALL DEVICES (CU M/HR)
60 C      CVOL=CABIN VOL (CU M)=DD(1,9)
61 C      SMNTC=SUM OF MASS NET TO CABIN(MG/HR)
62
63 C      CABIN VOL (CU M)
64 C      CVOL=DD(1,9)
65 C      CINITIAL (MG/CU M)
66 C      CINIT=CC(I,1)
67
68 C      EVALUATE SUM Q*REM EFF NET USING M.GEN IN DEVICES=0 (DD COL 19)
69 C      SET CABIN AVERAGE CONCENTRATION = TO ARBITRARY VALUE OF 100
70 C      AND INTERNAL GENERATION IN DEVICES =0 (NOT YET LOADED)
71 C      SET CABIN M.GEN=ARBITRARY VALUE OF 50 (DD(1,19))
72 C      SMGEN=SUM M.GEN IN ALL DEVICES +CABIN (MG/HR)
73 C      SMREM=SUM M. REMOVED BY ALL DEVICES (MG/HR)
74 C      CAV=CABIN AVERAGE CONCENTRATION (MG/CU M)
75 C      CAV=100
76 C      DD(1,19)=50
77
78 C      CALL CALCM(DD,NROW,NCOL,CAV,SMGEN,SMREM,TN,LIN2)
79 C      SQEFFN=SUM OF Q*REMOVAL EFF NET (MG/HR)
80 C      SQEFFN=SMREM/CAV
81
82 C
83 C      LOAD INTERNAL GENERATION FOR ALL DEV+CABIN FROM CDI INTO DD COL 19
84 C      CALL LDIGEN(I,DD,NROW,NCOL,CDI,NROW2,NCOL2,LIN2)
85 C
86 C      EVALUATE SUM OF M.NET TO CABIN=M.GEN CABIN+SUM M.GEN ALL DEVICES -
87 C      SUM M.REM ALL DEVICES
88 C      SMNTC=SUM M.NET TO CABIN=AMT GEN WHICH GETS TO THE CABIN DIRECTLY
89 C      SET C CABIN AV=0
90 C      CAV=0
91 C      GET SUM MASS GEN CABIN+ INTERNAL DEVICES AND SUM MASS REMOVED ALL
92 C      DEVICES FROM SUBROUTINE-SINCE CABIN C=0 NO CABIN CONT WILL BE REM
93 C      CALL CALCM(DD,NROW,NCOL,CAV,SMGEN,SMREM,TN,LIN2)
94 C      SMNTC=SMGEN-SMREM
95 C      NOTE:SMNTC IS ALSO PUT IN DD(1,21) BY CALCM
96 C
97 C      GET CALCULATED CABIN EQUILIBRIUM CONCENTRATION (CAVCLC) (MG/CU M)

```

```

98
99 C
100     IF(SQEFFN.LT.1E-10) THEN
101         IF(CVOL.EQ.0) THEN
102             CFINAL=1E10
103         ELSE
104             CFINAL=CINIT+(TN-TN1)*SMGEN/CVOL
105         ENDIF
106         CAVCLC=(CINIT+CFINAL)/2
107         CEQLIB=1E10
108     ELSE
109         CEQLIB=SMNTC/SQEFFN
110 C     CALCULATE CAVCLC AND CFINAL FROM SUBROUTINE
111         CALL PCAVCF(TN,TN1,CEQLIB,SQEFFN,CVOL,SMNTC,CINIT,
112 + CAVCLC,CFINAL)
113     ENDIF
114 C USING CAV CALC EVALUATE M.REM FOR CABIN + DEVICES AND PUT
115 C IN MAT DD COL 21
116     CAV=CAVCLC
117     CALL CALCM(DD,NROW,NCOL,CAV,SMGEN,SMREM,TN,LIN2)
118 C
119     RETURN
120 C END OF SUBROUTINE MASBAL
121 C ***** END OF SUBROUTINE MASBAL *****
122     END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C *****
2 C * SUBROUTINE MCALC *
3 C * MAIN CALCULATION LOOP SUBROUTINE FOR 1 TIME INCREMENT *
4 C * FOR ALL CONTAMINANTS ONE AT A TIME *
5 C * BASED ON SUM MASS REM LAST INCR, FOR EACH CONT *
6 C * CALCULATE NEW REMOVAL EFF, CAV CALC CABIN, *
7 C * CEQUILIB, CFINAL, & M.REMOVED ALL DEVICES-PUT IN MAT CC *
8 C *****
9 SUBROUTINE MCALC (I,TN,TN1,DD,NROW,NCOL,
10 +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,LIN,LIN2,
11 +PRTSW2,PRTSW3,PRTSW4,IMSGDN)
12 INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2,PRTSW2,PRTSW3,PRTSW4,KK
13 CHARACTER NN(NROW1)*30
14 REAL DD(NROW,NCOL)
15 REAL CC(NROW1,NCOL1)
16 REAL CDI(NROW2,NCOL2)
17 C SUBROUTINES REQUIRED:
18 C PREDCT=PREDICT CAV BASED ON M.GEN OF THIS INCR & REM EFF OF LST INC
19 C CONVRG=CALC CAV CALC,CEQ,CFINAL,M.REM,REM EFF
20 C CROUT=PRINT TEST VALUES OF MAT CC
21 C RROUT=PRINT TEST VALUES OF MAT DD
22 C
23 C INPUTS:
24 C FROM MAIN PROG
25 C I=CONTAMINANT NO.
26 C TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
27 C DD,NROW,NCOL=NAME & DIM OF MAT DD
28 C CC,NROW1,NCOL1=NAME & DIM OF MAT CC
29 C CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
30 C NN=NAME OF MAT NN
31 C LIN=NUMBER OF CONTAMINANTS IN MAT NN & CDI
32 C LIN2=NO. DEVICES IN MAT DD
33 C FROM PREDCT
34 C CAVPRD=PRED CABIN CONT CONC (MG/CU M)
35 C FROM CONVRG
36 C CAVCLC=CALC CABIN CONT CONC (MG/CU M)
37 C CFINAL=FINAL INCR CABIN CONT CONC(MG/CUM)
38 C KK=COUNTER FOR CONVERGENCE
39 C CEQLIB=EQUILIBRIUM CABIN CONT CONC(MG/CU M)
40 C IMSGDN=DEVICE NO FOR MESSAGE AND TEXT PRINTOUT OUTPUT
41 C OUTPUTS:
42 C TO PREDCT
43 C I=CONTAMINANT NO.
44 C TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
45 C DD,NROW,NCOL=NAME & DIM OF MAT DD
46 C CC,NROW1,NCOL1=NAME & DIM OF MAT CC
47 C CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI

```

```

48 C     LIN=NO. OF CONTAMINANTS IN MAT CDI
49 C     LIN2=NO. DEVICES IN MAT DD
50 C     TO CONVRG
51 C     I=CONTAMINANT NO.
52 C     TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
53 C     DD,NROW,NCOL=NAME & DIM OF MAT DD
54 C     CC,NROW1,NCOL1=NAME & DIM OF MAT CC
55 C     CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
56 C     LIN=NO. OF CONTAMINANTS IN MAT CDI
57 C     LIN2=NO. DEVICES IN MAT DD
58 C     CAVPRD=PREDICTED CABIN CONC FOR INCREMENT (MG/CU M)
59 C     TO MAT CC
60 C     PUTS CAVCLC,CEQLIV,&CFINAL IN CC(I,2-3 &4)
61 C     PUTS REM EFF FROM DD COL20 IN CC(I,7-10-13 ETC)
62 C     PUTS M.REM FOR EACH DEV FROM DD COL21 IN CC(I,6-9-12 ETC)
63 C     PUTS SUM MASS REM FOR EACH DEV IN CC(I,8-11-14 ETC)
64 C
65 C
66 C     BEGIN LOOP FOR EACH CONTAMINANT FOR EACH TIME INCREMENT
67 C     DO 100 I=1,LIN
68 C     CALC CAV PRED CABIN FOR CONT BASED ON REM EFF OF LAST INCREMENT
69 C     AND GENERATION RATES OF THIS INCREMENT
70 C     CALL PREDCT(I,TN,TN1,CAVPRD,DD,NROW,NCOL,
71 C     +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,LIN,LIN2,NN)
72 C
73 C
74 C     CONVERGE UNTIL CCALC=CPRED
75 C     CALL CONVRG(I,TN,TN1,CAVPRD,DD,NROW,NCOL,
76 C     +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,KK,LIN,
77 C     +LIN2,NN,PRTSW3,IMSGDN)
78 C
79 C     IF KK>20 THEN BEGIN 1/20 TIME INCREMENT CONVERGENCE ROUTINE
80 C     IF (KK.GT.20) THEN
81 C     KK=1
82 C     BEGIN 1/10 INCREMENT CONVERGENCE ROUTINE
83 C     NEW INCREMENT INITIAL TIME (HRS)
84 C     TN1NEW=TN1
85 C     NEW TIME INCREMENT (HRS)
86 C     BINEW=(TN-TN1)/20
87 C     BEGIN LOOP FOR 1/20 INCREMENT SIZE TIME INCREMENT
88 C     NEW INCREMENT FINAL TIME (HRS)
89 C     200   TNNNEW=TN1NEW+BINEW
90 C
91 C     ZERO MAT DD COL 17-21
92 C     CALL PRAFIL(DD,NROW,NCOL,17,21)
93 C     LOAD EFFICIENCY FROM LAST INCREMENT INTO MAT DD COL 20
94 C     CALL LODEFF(I,DD,NROW,NCOL,CC,NROW1,NCOL1,LIN2)
95 C     CALC CAV PRED CABIN FOR CONT BASED ON REM EFF OF LAST INCREMENT
96 C     AND GENERATION RATES OF THIS INCREMENT
97 C     CALL PREDCT(I,TNNNEW,TN1NEW,CAVPRD,DD,NROW,NCOL,

```

```

98      + CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,LIN,
99      + LIN2,NN)
100 C
101 C      CONVERGE UNTIL CCALC=CPRED
102      CALL CONVRG(I,TNNEW,TN1NEW,CAVPRD,DD,NROW,NCOL,
103      + CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,KK,LIN,
104      + LIN2,NN,PRTSW3,IMSGDN)
105 C
106 C      IF KK>20 THEN PRINT CONVERGENCE WARNING
107      IF (KK.GT.20) THEN
108          OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
109          WRITE(IMSGDN,*) 'WARNING: CALCULATION DID NOT CONVERGE FOR'
110          WRITE(IMSGDN,*) '          FULL AND 1/20 INCREMENT ROUTINES'
111          WRITE(IMSGDN,50) I,NN(I),TN1NEW,TNNEW
112      050      FORMAT (1X,'CONT NO.= ',I4,2X,A,/,1X,
113      +          'FOR INCREMENT INIT & FINAL TIMES= ',F8.2,F8.2)
114          CLOSE(IMSGDN)
115      ENDIF
116
117 C      FILL MAT CC WITH RESULTS
118 C      PUT CAVCLC,CEQLIB, AND CFINAL IN CC
119          CC(I,2)=CAVCLC
120          CC(I,3)=CEQLIB
121          CC(I,4)=CFINAL
122 C      PUT REM EFF FROM LAST ITER DD COL 20 IN CC(I,7-10-13ETC)
123          K=7
124          DO 102 J=2,LIN2
125              CC(I,K)=DD(J,20)
126              K=K+3
127      102      CONTINUE
128
129 C      TAKE CABIN M.REM(MG/HR) FROM DD(1,21) & PUT IN MAT CC(I,5)
130          CC(I,5)=DD(1,21)
131 C      TAKE M.REM FROM DD COL 21 & PUT IN CC(I,8-11-14ETC)
132          K=8
133          DO 103 J=2,LIN2
134              CC(I,K)=DD(J,21)
135              K=K+3
136      103      CONTINUE
137
138 C      CALCULATE SUM OF MASS REMOVED IN DEVICES + CABIN TO DATE AND
139 C      PUT IN CC(I,6-9-12ETC)
140          K=5
141          DO 104 J=1,LIN2
142              CC(I,K+1)=CC(I,K+1)+CC(I,K)*(TNNEW-TN1NEW)
143              K=K+3
144      104      CONTINUE
145 C
146 C      IF PRTSW2=1 THEN PRINT MAT CC INFO FOR THIS CONTAMINANT
147          IF (PRTSW2.EQ.1) THEN

```

```

148          OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
149          WRITE(IMSGDN,*)'NINC,TN,TN1 ',NINC,TN,TN1
150          WRITE(IMSGDN,*)'PRINTOUT FOR ONE CONT INSIDE 1/20 INCR
151 + LOOP OF MCALE'
152          WRITE(IMSGDN,*)'INFO FROM MAT CC'
153          CLOSE(IMSGDN)
154          CALL CROUT(NN,CC,NROW1,NCOL1,1,NCOL1,LIN,I,I,IMSGDN)
155          ENDIF
156 C
157
158 C          REPEAT LOOP FOR 1/20 INCREMENT IF END OF 1/20 INCREMENT TIME
159 C          (TNNEW) IS < THAN END OF LARGER TIME INCR (TN)
160 C          ELSE IF TNNEW>=TN, END 1/20 ENCR CONV & PRINT ANSWERS+REPEAT
161 C          FOR ANOTHER CONTAMINANT
162
163          IF (TNNEW.LT.TN) THEN
164 C          RESET FOR ANOTHER 1/20 TIME INCREMENT
165          TN1NEW=TNNEW
166          CC(I,1)=CC(I,4)
167          GO TO 200
168          ELSE
169 C          END 1/10 INCR CONV ROUTINE-REPEAT FOR ANOTHER CONT
170          GOTO 100
171          ENDIF
172
173 C
174 C          END OF CONVERGENCE ROUTINE
175          ENDIF
176
177 C          CALC SUM MASS REMOVED & FILL MAT CC WITH RESULTS
178 C          PUT CAVCLC,CEQLIB, AND CFINAL IN CC
179          CC(I,2)=CAVCLC
180          CC(I,3)=CEQLIB
181          CC(I,4)=CFINAL
182 C          PUT REM EFF FROM LAST ITER DD COL 20 IN CC(I,7-10-13ETC)
183          K=7
184          DO 302 J=2,LIN2
185          CC(I,K)=DD(J,20)
186          K=K+3
187          302 CONTINUE
188
189 C          TAKE CABIN M.REM(MG/HR) FROM DD(1,21) & PUT IN MAT CC(I,5)
190          CC(I,5)=DD(1,21)
191 C          TAKE M.REM FROM DD COL 21 & PUT IN CC(I,8-11-14ETC)
192          K=8
193          DO 303 J=2,LIN2
194          CC(I,K)=DD(J,21)
195          K=K+3
196          303 CONTINUE
197

```

```

198 C      CALCULATE SUM OF MASS REMOVED IN DEVICES + CABIN TO DATE AND PUT
199 C      IN CC(I,6-9-12ETC)
200      K=5
201      DO 304 J=1,LIN2
202          CC(I,K+1)=CC(I,K+1)+CC(I,K)*(TN-TN1)
203          K=K+3
204 304 CONTINUE
205 C
206
207 C      IF PRTSW4=1 THEN PRINT MAT DD+MAT CC INFO FOR THIS CONTAMINANT
208      IF (PRTSW4.EQ.1) THEN
209          OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
210          WRITE(IMSGDN,*)'PRINTOUT FOR ONE CONT AT END OF MCALC'
211          WRITE(IMSGDN,*)'INFO FROM MAT CC'
212          CLOSE(IMSGDN)
213          CALL CROUT(NN,CC,NROW1,NCOL1,1,NCOL1,LIN,I,I,IMSGDN)
214          OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
215          WRITE(IMSGDN,*)'INFO FROM MAT DD'
216          CLOSE(IMSGDN)
217          CALL RROUT(DD,NROW,NCOL,1,NCOL,LIN2,IMSGDN)
218      ENDIF
219 C
220 C      END OF I LOOP FOR EACH CONTAMINANT
221 C
222 100 CONTINUE
223      RETURN
224 C      ***** END OF SUBROUTINE MCALC *****
225      END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```



```

1 C      *****
2 C      *              SUBROUTINE PCAVCF              *
3 C      * SUBROUTINE TO PREDICT INCREMENT CALCULATED AVERAGE EFF      *
4 C      * (C AV CALC), FINAL EFF (CFINAL) & CABIN CONTAMINANT      *
5 C      * CONCENTRATION      *
6 C      *****
7
8      SUBROUTINE PCAVCF(TN,TN1,SCEQLIB,SQEFFN,CVOL,SMNTC,CINIT,
9      + CAVCLC,SCFINAL)
10     DOUBLE PRECISION EXPON,CEQLIB,CFINAL
11     CEQLIB=DBLE(SCEQLIB)
12
13 C SUBROUTINES REQUIRED: NONE
14 C
15 C INPUTS:
16 C TN,TN1=INITIAL & FINAL INCREMENT TIME (HRS)
17 C SCEQLIB(CEQLIB)=EQUILIBRIUM CABIN CONC (MG/CUM)
18 C SQEFFN=SUM Q*REMOVAL EFF NET (MG/HR)
19 C CVOL=CABIN VOLUME (CU M)
20 C SMNTC=SUM MASS CONT NET TO CABIN (MG/HR)
21 C CINIT=INITIAL INCREMENT CONT CONC (MG/CU M)
22 C OUTPUTS:
23 C SCAVCLC(CAVCLC)=CALC AVERAGE CABIN CONC (MG/CU M)
24 C SCFINAL(CFFINAL)=FINAL INCREMENT CONC (MG/CU M)
25 C
26     IF(CVOL.LE.0) THEN
27         CAVCLC=CEQLIB
28         CFFINAL=CEQLIB
29         GOTO 99
30     ENDIF
31 C     CALCULATION FOR CFINAL
32     EXPON=(TN-TN1)*SQEFFN/CVOL
33     IF(ABS(EXPON).GT.50) THEN
34         CAVCLC=CEQLIB
35         CFFINAL=CEQLIB
36         GOTO 99
37     ENDIF
38     IF(ABS(EXPON).LT.1E-6) THEN
39         CFFINAL=CINIT+SMNTC*(TN-TN1)/CVOL
40         CAVCLC=(CINIT+CFFINAL)/2
41         CEQLIB=1E10
42         GOTO 99
43     ELSE
44         CFFINAL=CINIT+(SMNTC/SQEFFN-CINIT)*(1-EXP(-EXPON))
45     ENDIF
46 C     CALCULATION FOR C AVERAGE CALC
47     IF ((CINIT.EQ.CFINAL).OR.(CFFINAL.EQ.CEQLIB)) THEN

```

```

48      CAVCLC=CFINAL
49      GOTO 99
50      ENDIF
51      IF ((CEQLIB-CINIT)/(CEQLIB-CFINAL).LT.1E-6) THEN
52          CAVCLC=(CINIT+CFINAL)/2
53      ELSE
54          CAVCLC=CEQLIB-(CFINAL-CINIT)/LOG((CEQLIB-CINIT)/
55      + (CEQLIB -CFINAL))
56      ENDIF
57 099 CONTINUE
58      SCFINAL=REAL(CFINAL)
59      RETURN
60 C      ***** END OF SUBROUTINE PCAVCF *****
61      END
62 C

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C      *****
2 C      *                      SUBROUTINE  PCSET                      *
3 C      *      SUBROUTINE  FOR PRECALCULATION SETUP ROUTINE          *
4 C      *      FOR ALL CONTAMINANTS ONE AT A TIME                    *
5 C      *      CALL EFF SUBROUTINES FOR DEVICES; GET CAV CABIN PRED,  *
6 C      *      CEQUILIB, CFINAL, & M.REMOVED ALL DEVICES-PUT IN MAT CC *
7 C      *****
8      SUBROUTINE PCSET(TN1,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,
9      +CDI,NROW2,NCOL2,LIN2,NN,PRTSW1,IMSGDN)
10     INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2,PRTSW1
11     CHARACTER NN(NROW1)*30
12     REAL DD(NROW,NCOL)
13     REAL CC(NROW1,NCOL1)
14     REAL CDI(NROW2,NCOL2)
15
16 C SUBROUTINES REQUIRED:
17 C   PRAFIL-ZERO MAT DD COL 17-21
18 C   CNRSUB-USING CAV=1E-20, FIND REMOVAL EFF AND PUT IN MAT DD COL 20
19 C   MASBAL-CALC CAV CALC PRED, CEQ, CFINAL, M.REM
20 C   CROUT-TEST PRINTOUT OF CONT INFO
21 C
22 C INPUTS:
23 C   FROM MAIN CALC LOOP
24 C   TN1=INCREMENT INITIAL TIME (HRS)
25 C   LIN=NO. OF CONT IN MAT CC AND NN
26 C   DD,NROW,NCOL=NAME & SIZE OF MAT DD
27 C   CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
28 C   CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
29 C   LIN2=NO. DEVICES IN MAT DD
30 C   NN=NAME OF MAT NN
31 C   PRTSW1=PRINTSWITCH WHICH CONTROLS TEST PRINTOUT
32 C   MSGDN=DEVICE NUMBER FOR MESSAGES AND TEST PRINTOUT
33 C FROM PRAFIL
34 C   PUTS ZEROS IN MAT DD COL 17-21
35 C FROM CNRSUB
36 C   CNRSUB PUTS REM EFF(DEC) FOR EACH DEVICE IN MAT DD COL 20
37 C FROM MASBAL (PREDICTED VALUES)
38 C   CAVCLC=AVERAGE CABIN CONC (MG/CU M)
39 C   CFINAL=FINAL INCREMENT CONT CONC (MG/CU M)
40 C   CEQLIB=EQUILIBRIUM CONT CONC (MG/CU M)
41 C   M.REM FOR ALL DEVICES PLACED IN COL 21 OF MAT DD
42 C OUTPUTS
43 C   TO MAIN PROGRAM
44 C   PUT IN MAT CC
45 C   CAVPRD=PRED CABIN AV CONC (MG/CU M): =CC(I,2)
46 C   CEQLIB=EQUILIBRIUM CABIN CONT CONC (MG/CU M): =CC(I,3)
47 C   CFINAL=FINAL CABIN CONT CONC (MG/CU M): =CC(I,4)

```

```

48 C      PUTS REM EFF FROM DD COL 20 IN CC(I,7-10-13-16 ETC)
49 C      PUTS M.REM IN CC(I,5-8-11-14...)
50 C      TO PRAFIL
51 C      NAME AND SIZE OF MAT DD+FIRST & LAST COLUMN TO ZERO
52 C      TO CNRSUB
53 C      I=CONT NO.
54 C      TN,TN1=CONT INCREMENT FINAL,INIT TIME (HRS)
55 C      CAVPRD=CABIN AVERAGE CONT CONC (MG/CU M)
56 C      DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
57 C      CC,NROW1,NCOL1=NAME & DIM OF MAT CC
58 C      CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
59 C      LIN2=NO. ACTIVE DEVICES IN MAT DD
60 C      TO MASBAL
61 C      I=CONT NO.
62 C      TN,TN1=CONT INCREMENT FINAL,INIT TIME (HRS)
63 C      DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
64 C      CALCLC=CALC CABIN AV CONC (MG/CU M)
65 C      CC,NROW1,NCOL1=NAME & DIM OF MAT CC
66 C      CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
67 C      CFINAL=CABIN FINAL CONCENTRATION (MG/CU M)
68 C      CEQLIB=CABIN EQUILIBRIUM CONCENTRATION (MG/CU M)
69 C      LIN=NO. OF CONTAMINANTS IN MAT CDI
70 C      LIN2=NO. ACTIVE DEVICES IN MAT DD
71 C      TO RROUT
72 C      MATRIX NAME,#ROWS,#COLS,FIRST & LAST COL TO PRINT,#LINES TO PRINT
73
74 C      BASIC TIME INCREMENT (HRS)
75 C      BINC=DD(1,11)
76 C      SET FINAL INCREMENT TIME (HRS)
77 C      TN=0.1*BINC/24
78 C      BEGIN LOOP FOR EACH CONTAMINANT - ONE AT A TIME
79 C      CALCULATE REM EFF FOR EACH DEVICE, GET M.REM, CAV CABIN CALC
80 C      CEQLIB, CFINAL-PUT IN MAT CC
81 C      DO 100 I=1,LIN
82 C          ZERO MAT DD COL 17 TO 21
83 C          CALL PRAFIL(DD,NROW,NCOL,17,21)
84 C          SET CAVPRD = MINIMUM VALUE TO ALLOW COMPUTATION
85 C          CAVPRD=1E-20
86 C          DD(J,22)=CAVPRD
87 C          CALC REMOVAL EFFICIENCY (THROUGH EFF CALLING SUBROUTINE)
88 C          THIS STORES REM EFF IN MAT DD COL 20 FOR EACH DEVICE
89 C          CALL CNRSUB(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
90 C      +      CDI,NROW2,NCOL2,LIN2)
91 C          CALL MASS BALANCE-GET CAVPRD(=CAVCLC IN MASBAL),CEQ,CFINAL,M.REM
92 C          CALL MASBAL(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
93 C      +      CAVCLC,CDI,NROW2,NCOL2,CFINAL,CEQLIB,LIN,LIN2)
94 C          CAVPRD=CAVCLC
95 C          PUT CAVPRD,CEQLIB,& CFINAL IN MAT CC
96 C          CC(I,2)=CAVPRD
97 C          CC(I,3)=CEQLIB

```

```

108      CC(I,4)=CFINAL
109 C      GET REM EFF FROM DD COL 20 AND PUT IN CC(I,7-10-13 ETC)
110      K=7
111      DO 101 J=2,LIN2
112      CC(I,K)=DD(J,20)
113      K=K+3
114 101    CONTINUE
115 C      TAKE M.REMOVED FROM MAT DD COL 21 AND PUT IN MAT CC(I,5-8-...)
116 C      CABIN REMOVAL RATE
117      CC(I,5)=DD(1,21)
118 C      DEVICE 2-15 REMOVAL RATE
119      K=8
120      DO 102 J=2,LIN2
121      CC(I,K)=DD(J,21)
122      K=K+3
123 102    CONTINUE
124 C      IF PRTSW1=1 THEN PRINT MAT DD+MAT CC INFO FOR THIS CONTAMINANT
125      IF (PRTSW1.EQ.1) THEN
126      OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
127      WRITE(IMSGDN,*) 'PRINTOUT FOR ONE CONT AT END OF PCSET'
128      WRITE(IMSGDN,*) 'INFO FROM MAT CC'
129      CLOSE (IMSGDN)
130      CALL CROUT(NN,CC,NROW1,NCOL1,1,NCOL1,LIN,I,I,IMSGDN)
131      OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
132      WRITE(IMSGDN,*) 'INFO FROM MAT DD'
133      CLOSE (IMSGDN)
134      CALL RROUT(DD,NROW,NCOL,1,NCOL,LIN2,IMSGDN)
135      ENDIF
136 C      100 CONTINUE
137 C      RETURN
138 C      ***** END OF SUBROUTINE PCSET *****
139 C      END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C *****
2 C *
3 C *      SUBROUTINE PRAFIL
4 C *      SUBROUTINE TO FILL ADJUSTABLE SIZE REAL ARRAY WITH ZEROS
5 C *      PARTIAL FILL-FROM COL FSTCOL TO COL LSTCOL
6 C *****
7 SUBROUTINE PRAFIL(X,NROW,NCOL,FSTCOL,LSTCOL)
8 INTEGER NCOL,NROW,FSTCOL,LSTCOL
9 REAL X(NROW,NCOL)
10 C INPUTS:
11 C X,NROW,NCOL=NAME AND DIMENSIONS OF MATRIX X
12 C FSTCOL,LSTCOL=FIRST AND LAST COLUMN TO FILL WITH ZEROS
13
14 IF(FSTCOL.GT.NCOL) FSTCOL=NCOL
15 IF(LSTCOL.GT.NCOL) LSTCOL=NCOL
16 IF(FSTCOL.GT.LSTCOL) FSTCOL=LSTCOL
17 DO 110 I=1,NROW
18 DO 100 J=FSTCOL,LSTCOL
19 X(I,J)=0.0
20 100 CONTINUE
21 110 CONTINUE
22 RETURN
23 C ***** END OF SUBROUTINE PRAFIL *****
24 END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS IN COMPILATION : 0

```

```

1 C  FILE PRFANS
2 C  *****
3 C  *      SUBROUTINE PRCDA      *
4 C  *  PROGRAM TO PRINT ANSWERS FOR CONCENTRATION DATA  *
5 C  *****
6
7 C  NOTES:(1)FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
8 C  (2)IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED
9
10     SUBROUTINE PRCDA(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
11 +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR,
12 +PRTSW8,PRTSW9,FCPLOT,IDEVN1)
13 C  SUBROUTINES REQUIRED:
14 C  HDG1,HDG2,HDG3,HDG4
15
16 C  TN,TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
17 C  LIN=TOTAL NUMBER OF CONTAMINANTS
18 C  CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
19 C  CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
20 C  NN=NAME OF MAT NN
21 C  IDEVNO=DEVICE NUMBER FOR OUTPUT
22 C  NINC=TIME INCREMENT NUMBER
23 C  =0 THEN PRINT HDG3 WITH PCALC
24 C  =-1 THEN PRINT HDG3 WITH FINAL
25 C  ELSE PRINT HDG2 WITH INCREMENT NUMBER
26 C  IMONTH..IMINUTE=TIME AND DATE INFO
27 C  FNAME=FILE NAME OUTPUT DATA IS STORED ON
28 C  IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT
29
30     REAL CC(NROW1,NCOL1)
31     REAL CDI(NROW2,NCOL2)
32     CHARACTER CNAME*30,FNAME*24,ECHR*1,FCPLOT*24
33     CHARACTER NN(NROW1)*30
34     INTEGER PRTSW8,PRTSW9,NINC
35 C  ECHR=EXCEEDS MAC CHARACTER (Y OR N)
36     IF ((PRTSW8.EQ.1).OR.((PRTSW8.EQ.0).AND.(NINC.EQ.-1))) THEN
37 C  INCREMENT PAGE COUNTER BY ONE
38     IPGCTR=IPGCTR+1
39
40 C  START FIRST PAGE
41 C  PRINT FORM FEED
42     WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
43 020  FORMAT('1')
44 C  PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &4
45     WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
46 040  FORMAT(1X,'')
47     CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)

```

```

48      IF (NINC.EQ.0) THEN
49          CALL HDG3 (1,TN1,TN,IDEVNO)
50      ELSEIF (NINC.EQ.-1) THEN
51          CALL HDG3 (2,TN1,TN,IDEVNO)
52      ELSE
53          CALL HDG2 (NINC,TN1,TN,IDEVNO)
54      ENDIF
55      CALL HDG4 (IDEVNO)
56 C    PRINT ANOTHER BLANK LINE
57          WRITE (IDEVNO,40,IOSTAT=IOVAL,ERR=900)
58
59 C    BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
60      DO 100 I=1,LIN
61
62 C          CNAME=CONTAMINANT NAME
63          CNAME=NN(I)
64 C          FCONC=FINAL CONT CONCENTRATION (MG CU M)
65          FCONC=CC(I,4)
66 C          RMAC=MAXIMUM ALLOWABLE CONCENTRATION (MG/CU M)
67          RMAC=CDI(I,9)
68
69 C          IF CABIN CONC>MAC PRINT 'Y' OTHERWISE PRINT 'N'
70          IF (FCONC.GT.RMAC) THEN
71              ECHR='Y'
72          ELSE
73              ECHR='N'
74          ENDIF
75 C
76 C          PRINT 56 LINES OF DATA AND THEN START NEW PAGE
77          WRITE (IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,CNAME,FCONC,RMAC,ECHR
78      010  FORMAT(1X,I4,1X,A,1X,G11.4,1X,G11.4,5X,A)
79
80 C          CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
81          IF (INT (REAL(I)/56).EQ.REAL(I)/56) THEN
82              IPGCTR=IPGCTR+1
83 C          START SUBSEQUENT PAGES
84 C          PRINT FORM FEED
85              WRITE (IDEVNO,50,IOSTAT=IOVAL,ERR=900)
86      050  FORMAT('1')
87 C          PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &4
88          WRITE (IDEVNO,30,IOSTAT=IOVAL,ERR=900)
89      030  FORMAT(1X,'')
90          CALL HDG1 (IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
91          IF (NINC.EQ.0) THEN
92              CALL HDG3 (1,TN1,TN,IDEVNO)
93          ELSEIF (NINC.EQ.-1) THEN
94              CALL HDG3 (2,TN1,TN,IDEVNO)
95          ELSE
96              CALL HDG2 (NINC,TN1,TN,IDEVNO)

```



```

97         ENDIF
98         CALL HDG4 (IDEVNO)
99 C        PRINT ANOTHER BLANK LINE
100        WRITE (IDEVNO, 30, IOSTAT=IOVAL, ERR=900)
101
102        ENDIF
103 100 CONTINUE
104        ENDIF
105 C
106 C        ***** WRITE CONCENTRATION DATA TO A PLOT FILE *****
107        IF (NINC.NE.-1) THEN
108        IF ((PRTSW9.EQ.1).OR.(PRTSW9.EQ.3)) THEN
109            DO 120 I=1,LIN,300
110                IS=I
111                IE=I+299
112                IF (IE.GT.LIN) IE=LIN
113                WRITE (IDEVN1,110,IOSTAT=IOVAL,ERR=900) TN1,TN,
114                +      (CC(J,4),J=IS,IE)
115 110        FORMAT (T2,2(F8.3,1X),300(G11.4, :,1X))
116 120 CONTINUE
117        ENDIF
118        ENDIF
119        GO TO 999
120 900 WRITE(*,*) 'IO ERROR IN PRCD= ',IOVAL
121 999 RETURN
122        END

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS IN PROGRAM UNIT: 0

```

123
124 C        *****
125 C        *          SUBROUTINE PRREM1          *
126 C        * PROGRAM TO PRINT ANSWERS-RATE OF CONTAMINAMT REMOVAL (MG/HR) *
127 C        * SHEET 1          *
128 C        *****
129
130 C        NOTES: (1) FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
131 C              (2) IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED
132
133        SUBROUTINE PRREM1 (TN, TN1, LIN, CC, NROW1, NCOL1, CDI, NROW2, NCOL2, NN,
134        +IDEVNO, NINC, IMONTH, IDAY, IYEAR, IHOURL, IMINUTE, FNAME, IOVAL, IPGCTR)
135
136 C        SUBROUTINES REQUIRED:
137 C        HDG1, HDG2, HDG3, HDG8
138
139 C        TN, TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
140 C        LIN=TOTAL NUMBER OF CONTAMINANTS
141 C        CC, NROW1, NCOL1=NAME & SIZE OF MAT CC
142 C        CDI, NROW2, NCOL2=NAME & SIZE OF MAT CDI
143 C        NN=NAME OF MAT NN

```

```

144 C      IDEVNO=DEVICE NUMBER FOR OUTPUT
145 C      NINC=TIME INCREMENT NUMBER
146 C      =0 THEN PRINT HDG3 WITH PCALC
147 C      =-1 THEN PRINT HDG3 WITH FINAL
148 C      ELSE PRINT HDG2 WITH INCREMENT NUMBER
149 C      IMONTH..IMINUTE=TIME AND DATE INFO
150 C      FNAME=FILE NAME OUTPUT DATA IS STORED ON
151 C      IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT
152
153      REAL CC(NROW1,NCOL1)
154      REAL CDI(NROW2,NCOL2)
155      CHARACTER CNAME*30,FNAME*24
156      CHARACTER NN(NROW1)*30
157
158 C      INCREMENT PAGE COUNTER BY ONE
159      IPGCTR=IPGCTR+1
160
161 C      START FIRST PAGE
162 C      PRINT FORM FEED
163      WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
164      020  FORMAT('1')
165 C      PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &8
166      WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
167      040  FORMAT(1X,' ')
168      CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
169      IF(NINC.EQ.0) THEN
170          CALL HDG3(1,TN1,TN,IDEVNO)
171      ELSEIF(NINC.EQ.-1) THEN
172          CALL HDG3(2,TN1,TN,IDEVNO)
173      ELSE
174          CALL HDG2(NINC,TN1,TN,IDEVNO)
175      ENDIF
176      CALL HDG8(IDEVNO)
177 C      PRINT ANOTHER BLANK LINE
178      WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
179
180 C      BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
181      DO 100 I=1,LIN
182
183 C          PRINT 56 LINES OF DATA AND THEN START NEW PAGE
184          WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,NN(I),
185          +CC(I,5),CC(I,8),CC(I,11),CC(I,14),CC(I,17),CC(I,20),CC(I,23),
186          +CC(I,26)
187      010  FORMAT(1X,I4,1X,A,8(1X,G11.4))
188
189 C      CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
190      IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
191          IPGCTR=IPGCTR+1
192 C      START SUBSEQUENT PAGES
193 C      PRINT FORM FEED

```

```

194          WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)
195      050      FORMAT('1')
196 C          PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 & 8
197          WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
198      030      FORMAT(1X,'')
199          CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
200          IF(NINC.EQ.0) THEN
201              CALL HDG3(1,TN1,TN,IDEVNO)
202          ELSEIF(NINC.EQ.-1) THEN
203              CALL HDG3(2,TN1,TN,IDEVNO)
204          ELSE
205              CALL HDG2(NINC,TN1,TN,IDEVNO)
206          ENDIF
207          CALL HDG8(IDEVNO)
208 C          PRINT ANOTHER BLANK LINE
209          WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
210
211          ENDIF
212      100 CONTINUE
213
214          GO TO 999
215      900 WRITE(*,*) 'IO ERROR IN PRREM1= ',IOVAL
216      999 RETURN
217          END

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS IN PROGRAM UNIT: 0

```

218
219 C          *****
220 C          *          SUBROUTINE PRREM2          *
221 C          * PROGRAM TO PRINT ANSWERS-RATE OF CONTAMINANT REMOVAL (MG/HR) *
222 C          * SHEET 2          *
223 C          *****
224
225 C          NOTES: (1) FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
226 C                  (2) IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED
227
228          SUBROUTINE PRREM2(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
229      +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR) 230
231 C          SUBROUTINES REQUIRED:
232 C          HDG1,HDG2,HDG3,HDG9
233
234 C          TN,TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
235 C          LIN=TOTAL NUMBER OF CONTAMINANTS
236 C          CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
237 C          CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
238 C          NN=NAME OF MAT NN
239 C          IDEVNO=DEVICE NUMBER FOR OUTPUT

```

```

240 C      NINC=TIME INCREMENT NUMBER
241 C      =0 THEN PRINT HDG3 WITH PCALC
242 C      =-1 THEN PRINT HDG3 WITH FINAL
243 C      ELSE PRINT HDG2 WITH INCREMENT NUMBER
244 C      IMONTH..IMINUTE=TIME AND DATE INFO
245 C      FNAME=FILE NAME OUTPUT DATA IS STORED ON
246 C      IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT
247
248      REAL CC(NROW1,NCOL1)
249      REAL CDI(NROW2,NCOL2)
250      CHARACTER CNAME*30,FNAME*24
251      CHARACTER NN(NROW1)*30
252
253 C      INCREMENT PAGE COUNTER BY ONE
254      IPGCTR=IPGCTR+1
255
256 C      START FIRST PAGE
257 C      DON'T PRINT FORM FEED UNLESS NO. CONT > 20
258      IF(LIN.GT.20) THEN
259 C          PRINT FORM FEED
260          WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
261      020      FORMAT('1')
262      ENDIF
263 C      PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 & 9
264          WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
265      040      FORMAT(1X,' ')
266          CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
267          IF(NINC.EQ.0) THEN
268              CALL HDG3(1,TN1,TN,IDEVNO)
269          ELSEIF(NINC.EQ.-1) THEN
270              CALL HDG3(2,TN1,TN,IDEVNO)
271          ELSE
272              CALL HDG2(NINC,TN1,TN,IDEVNO)
273          ENDIF
274          CALL HDG9(IDEVNO)
275 C      PRINT ANOTHER BLANK LINE
276          WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
277
278 C      BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
279      DO 100 I=1,LIN
280
281 C          PRINT 56 LINES OF DATA AND THEN START NEW PAGE
282          WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,NN(I),
283      +CC(I,29),CC(I,32),CC(I,35),CC(I,38),CC(I,41),CC(I,44),CC(I,47)
284      010      FORMAT(1X,I4,1X,A,7(1X,G11.4))
285
286 C      CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
287          IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
288              IPGCTR=IPGCTR+1
289 C              START SUBSEQUENT PAGES

```

```

290 C          PRINT FORM FEED
291          WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)
292      050      FORMAT('1')
293 C          PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &8
294          WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
295      030      FORMAT(1X,'')
296          CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
297          IF(NINC.EQ.0) THEN
298              CALL HDG3(1,TN1,TN,IDEVNO)
299          ELSEIF(NINC.EQ.-1) THEN
300              CALL HDG3(2,TN1,TN,IDEVNO)
301          ELSE
302              CALL HDG2(NINC,TN1,TN,IDEVNO)
303          ENDIF
304          CALL HDG9(IDEVNO)
305 C          PRINT ANOTHER BLANK LINE
306          WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
307
308          ENDIF
309      100 CONTINUE
310
311          GO TO 999
312      900 WRITE(*,*)'IO ERROR IN PRREM2= ',IOVAL
313      999 RETURN
314          END

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS IN PROGRAM UNIT: 0

```

315
316 C          *****
317 C          *          SUBROUTINE PRMAS1          *
318 C          * PROGRAM TO PRINT ANSWERS-SUM OF CONT REMOVED BY DEVICE (MG) *
319 C          * SHEET 1          *
320 C          *****
321
322 C          NOTES:(1)FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
323 C          (2)IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED
324
325          SUBROUTINE PRMAS1(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
326 +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)
327
328 C          SUBROUTINES REQUIRED:
329 C          HDG1,HDG2,HDG3,HDG5
330
331 C          TN,TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
332 C          LIN=TOTAL NUMBER OF CONTAMINANTS
333 C          CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
334 C          CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
335 C          NN=NAME OF MAT NN

```

```

336 C      IDEVNO=DEVICE NUMBER FOR OUTPUT
337 C      NINC=TIME INCREMENT NUMBER
338 C      =0 THEN PRINT HDG3 WITH PCALC
339 C      =-1 THEN PRINT HDG3 WITH FINAL
340 C      ELSE PRINT HDG2 WITH INCREMENT NUMBER
341 C      IMONTH..IMINUTE=TIME AND DATE INFO
342 C      FNAME=FILE NAME OUTPUT DATA IS STORED ON
343 C      IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT
344
345      REAL CC(NROW1,NCOL1)
346      REAL CDI(NROW2,NCOL2)
347      CHARACTER CNAME*30,FNAME*24
348      CHARACTER NN(NROW1)*30
349
350 C      INCREMENT PAGE COUNTER BY ONE
351      IPGCTR=IPGCTR+1
352
353 C      START FIRST PAGE
354 C      PRINT FORM FEED
355      WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
356 020  FORMAT('1')
357 C      PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &5
358      WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
359 040  FORMAT(1X,' ')
360      CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
361      IF(NINC.EQ.0) THEN
362          CALL HDG3(1,TN1,TN,IDEVNO)
363      ELSEIF(NINC.EQ.-1) THEN
364          CALL HDG3(2,TN1,TN,IDEVNO)
365      ELSE
366          CALL HDG2(NINC,TN1,TN,IDEVNO)
367      ENDIF
368      CALL HDG5(IDEVNO)
369 C      PRINT ANOTHER BLANK LINE
370      WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
371
372 C      BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
373      DO 100 I=1,LIN
374
375 C          PRINT 56 LINES OF DATA AND THEN START NEW PAGE
376          WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,NN(I),
377      +CC(I,6),CC(I,9),CC(I,12),CC(I,15),CC(I,18),CC(I,21),CC(I,24),
378      +CC(I,27)
379 010  FORMAT(1X,I4,1X,A,8(1X,G11.4))
380
381 C          CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
382          IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
383              IPGCTR=IPGCTR+1
384 C              START SUBSEQUENT PAGES
385 C              PRINT FORM FEED

```

```

386         WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)
387     050     FORMAT('1')
388 C     PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 & 8
389         WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
390     030     FORMAT(1X,'')
391         CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
392         IF(NINC.EQ.0) THEN
393             CALL HDG3(1,TN1,TN,IDEVNO)
394         ELSEIF(NINC.EQ.-1) THEN
395             CALL HDG3(2,TN1,TN,IDEVNO)
396         ELSE
397             CALL HDG2(NINC,TN1,TN,IDEVNO)
398         ENDIF
399         CALL HDG5(IDEVNO)
400 C     PRINT ANOTHER BLANK LINE
401         WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
402
403         ENDIF
404     100 CONTINUE
405
406         GO TO 999
407     900 WRITE(*,*)'IO ERROR IN PRMAS1= ',IOVAL
408     999 RETURN
409         END

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS IN PROGRAM UNIT: 0

```

410
411 C     *****
412 C     *          SUBROUTINE PRMAS2                      *
413 C     * PROGRAM TO PRINT ANSWERS-SUM OF CONT REMOVED BY DEVICE (MG) *
414 C     * SHEET 2                                          *
415 C     *****
416
417 C     NOTES: (1) FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
418 C            (2) IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED
419
420         SUBROUTINE PRMAS2(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
421 +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)
422
423 C     SUBROUTINES REQUIRED:
424 C         HDG1,HDG2,HDG3,HDG6
425
426 C     TN,TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
427 C     LIN=TOTAL NUMBER OF CONTAMINANTS
428 C     CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
429 C     CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
430 C     NN=NAME OF MAT NN
431 C     IDEVNO=DEVICE NUMBER FOR OUTPUT

```

```

432 C      NINC=TIME INCREMENT NUMBER
433 C      =0 THEN PRINT HDG3 WITH PCALC
434 C      =-1 THEN PRINT HDG3 WITH FINAL
435 C      ELSE PRINT HDG2 WITH INCREMENT NUMBER
436 C      IMONTH..IMINUTE=TIME AND DATE INFO
437 C      FNAME=FILE NAME OUTPUT DATA IS STORED ON
438 C      IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT
439
440      REAL CC(NROW1,NCOL1)
441      REAL CDI(NROW2,NCOL2)
442      CHARACTER CNAME*30,FNAME*24
443      CHARACTER NN(NROW1)*30
444
445 C      INCREMENT PAGE COUNTER BY ONE
446      IPGCTR=IPGCTR+1
447
448 C      START FIRST PAGE
449 C      DON'T PRINT FORM FEED UNLESS NO. CONT > 20
450      IF(LIN.GT.20) THEN
451 C          PRINT FORM FEED
452          WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
453 020      FORMAT('1')
454      ENDIF
455 C      PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &9
456          WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
457 040      FORMAT(1X,' ')
458          CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
459          IF(NINC.EQ.0) THEN
460              CALL HDG3(1,TN1,TN,IDEVNO)
461          ELSEIF(NINC.EQ.-1) THEN
462              CALL HDG3(2,TN1,TN,IDEVNO)
463          ELSE
464              CALL HDG2(NINC,TN1,TN,IDEVNO)
465          ENDIF
466          CALL HDG6(IDEVNO)
467 C      PRINT ANOTHER BLANK LINE
468          WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
469
470 C      BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
471      DO 100 I=1,LIN
472
473 C          PRINT 56 LINES OF DATA AND THEN START NEW PAGE
474          WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,NN(I),
475      +CC(I,30),CC(I,33),CC(I,36),CC(I,39),CC(I,42),CC(I,45),CC(I,48)
476 010      FORMAT(1X,I4,1X,A,7(1X,G11.4))
477
478 C      CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
479          IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
480              IPGCTR=IPGCTR+1
481 C              START SUBSEQUENT PAGES

```



```

482 C      PRINT FORM FEED
483          WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)
484 050      FORMAT('1')
485 C      PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &8
486          WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
487 030      FORMAT(1X,'')
488          CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
489          IF(NINC.EQ.0) THEN
490              CALL HDG3(1,TN1,TN,IDEVNO)
491          ELSEIF(NINC.EQ.-1) THEN
492              CALL HDG3(2,TN1,TN,IDEVNO)
493          ELSE
494              CALL HDG2(NINC,TN1,TN,IDEVNO)
495          ENDIF
496          CALL HDG6(IDEVNO)
497 C      PRINT ANOTHER BLANK LINE
498          WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
499
500          ENDIF
501 100 CONTINUE
502
503          GO TO 999
504 900 WRITE(*,*)'IO ERROR IN PRMAS2= ',IOVAL
505 999 RETURN
506      END

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS IN PROGRAM UNIT: 0

```

507
508 C      *****
509 C      *          SUBROUTINE PREFF          *
510 C      * PROGRAM TO PRINT ANSWERS-END OF INCREMENT REMOVAL EFF (DEC) *
511 C      *          *
512 C      *****
513
514 C      NOTES:(1)FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
515 C      (2)IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED
516
517      SUBROUTINE PREFF(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
518 +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR,
519 +PRTSW8,PRTSW9,IDEVN2)
520
521 C      SUBROUTINES REQUIRED:
522 C      HDG1,HDG2,HDG3,HDG7
523
524 C      TN,TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
525 C      LIN=TOTAL NUMBER OF CONTAMINANTS
526 C      CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
527 C      CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI

```

```

528 C      NN=NAME OF MAT NN
529 C      IDEVNO=DEVICE NUMBER FOR OUTPUT
530 C      NINC=TIME INCREMENT NUMBER
531 C      =0 THEN PRINT HDG3 WITH PCALC
532 C      =-1 THEN PRINT HDG3 WITH FINAL
533 C      ELSE PRINT HDG2 WITH INCREMENT NUMBER
534 C      IMONTH..IMINUTE=TIME AND DATE INFO
535 C      FNAME=FILE NAME OUTPUT DATA IS STORED ON
536 C      IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT
537
538      REAL CC(NROW1,NCOL1)
539      REAL CDI(NROW2,NCOL2)
540      CHARACTER CNAME*30,FNAME*24
541      CHARACTER NN(NROW1)*30
542      INTEGER PRTSW8,PRTSW9,IDEVN2,IDEVNO,I,J,K,H,NINC
543      IF ((PRTSW8.EQ.1).OR.((PRTSW8.EQ.0).AND.(NINC.EQ.-1))) THEN
544 C      INCREMENT PAGE COUNTER BY ONE
545      IPGCTR=IPGCTR+1
546
547 C      START FIRST PAGE
548 C      PRINT FORM FEED
549      WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
550 020  FORMAT('1')
551 C      PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &9
552      WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
553 040  FORMAT(1X,'')
554      CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
555      IF(NINC.EQ.0) THEN
556          CALL HDG3(1,TN1,TN,IDEVNO)
557      ELSEIF(NINC.EQ.-1) THEN
558          CALL HDG3(2,TN1,TN,IDEVNO)
559      ELSE
560          CALL HDG2(NINC,TN1,TN,IDEVNO)
561      ENDIF
562      CALL HDG7(IDEVNO)
563 C      PRINT ANOTHER BLANK LINE
564      WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
565
566 C      BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
567      DO 100 I=1,LIN
568
569 C          PRINT 56 LINES OF DATA AND THEN START NEW PAGE
570          WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,NN(I),
571      +CC(I,7),CC(I,10),CC(I,13),CC(I,16),CC(I,19),CC(I,22),CC(I,25),
572      +CC(I,28),CC(I,31),CC(I,34),CC(I,37),CC(I,40),CC(I,43),CC(I,46)
573 010  FORMAT(1X,I4,1X,A,14(1X,F5.3))
574
575 C      CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
576      IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
577          IPGCTR=IPGCTR+1

```

```

578 C          START SUBSEQUENT PAGES
579 C          PRINT FORM FEED
580             WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)
581 050         FORMAT('1')
582 C          PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 & 8
583             WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
584 030         FORMAT(1X,'')
585             CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
586             IF(NINC.EQ.0) THEN
587                 CALL HDG3(1,TN1,TN,IDEVNO)
588             ELSEIF(NINC.EQ.-1) THEN
589                 CALL HDG3(2,TN1,TN,IDEVNO)
590             ELSE
591                 CALL HDG2(NINC,TN1,TN,IDEVNO)
592             ENDIF
593             CALL HDG7(IDEVNO)
594 C          PRINT ANOTHER BLANK LINE
595             WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
596
597             ENDIF
598 100 CONTINUE
599             ENDIF
600 C          ***** WRITE DATA TO A FILE FOR PLOTTING *****
601             IF (NINC.NE.-1) THEN
602             IF ((PRTSW9.EQ.2).OR.(PRTSW9.EQ.3)) THEN
603                 DO 70 I=1,LIN,300
604                     IS=I
605                     IE=I+299
606                     IF (IE.GT.LIN) IE=LIN
607                     K=7
608                     DO 60 H=2,15
609                         WRITE (IDEVN2,55,IOSTAT=IOVAL,ERR=900) TN1,TN,H,
610                             +      (CC(J,K),J=IS,IE)
611 055                 FORMAT (T2,2(F8.2,1X),I2,1X,300(F5.3,:,1X))
612                     K=K+3
613 060                 CONTINUE
614 070                 CONTINUE
615             ENDIF
616             ENDIF
617             GO TO 999
618 900 WRITE(*,*)'IO ERROR IN PREF= ',IOVAL
619 999 RETURN
620             END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C *****
2 C * SUBROUTINE PREDCT *
3 C * BASED ON REMOVAL EFF & SUM MASS REMOVED OF LAST INCREMENT, *
4 C * AND M.GEN OF THIS INCREMENT, PREDICT CAV PRED *
5 C * (CEQUILIB, CFINAL, M.REM ARE ALSO CALC, BUT NOT NEEDED) *
6 C * WORKS FOR ONE CONT AT A TIME *
7 C *****
8
9 SUBROUTINE PREDCT(I,TN,TN1,CAVPRD,DD,NROW,NCOL,
10 +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,LIN,LIN2,NN)
11
12 INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2
13 CHARACTER NN(NROW1)*30
14 REAL DD(NROW,NCOL)
15 REAL CC(NROW1,NCOL1)
16 REAL CDI(NROW2,NCOL2)
17 C
18 C SUBROUTINES REQUIRED:
19 C PRAFIL-ZERO MAT DD COL 17-21
20 C LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
21 C MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
22 C
23 C INPUTS:
24 C FROM MCALC
25 C I=CONTAMINANT NO.
26 C TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
27 C DD,NROW,NCOL=NAME & DIM OF MAT DD
28 C CC,NROW1,NCOL1=NAME & DIM OF MAT CC
29 C CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
30 C LIN=NO. OF CONTAMINANTS IN MAT CDI
31 C LIN2=NO. DEVICES IN MAT DD
32 C NN=NAME OF MAT NN
33 C FROM LODEFF
34 C TAKES REM EFF FOR LAST INCR (IN MAT CC) AND PUTS IT IN
35 C MAT DD COL 20 (FOR ALL DEVICES)
36 C FROM MASBAL
37 C CAVCLC=CALC CABIN CONT CONC (MG/CU M)
38 C OUTPUTS:
39 C TO LODEFF
40 C I=CONTAMINANT LINE NUMBER IN MAT CC
41 C DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
42 C CC,NROW1,NCOL1=NAME & DIMENSIONS OF MAT CC
43 C LIN2=NO. OF DEVICES IN MAT DD
44 C TO MASBAL
45 C TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
46 C CNVERR=CONVERGENCE ERROR
47 C DD,NROW,NCOL=NAME & DIM OF MAT DD

```

```

48 C      CC,NROW1,NCOL1=NAME & DIM OF MAT CC
49 C      CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
50 C      LIN=NO. OF CONTAMINANTS IN MAT CDI
51 C      LIN2=NO. DEVICES IN MAT DD
52 C      CVOL=CABIN VOLUME (CU M)
53 C      TCABIN=CABIN TEMP (DEG K)
54 C      CINIT=INCR INIT CABIN CONT CONC (MG/CU M)=CC(I,1)
55 C      TO MCALC
56 C      CAVCLC
57 C
58 C      ZERO MAT DD COL 17-21
59 C      CALL PRAFIL(DD,NROW,NCOL,17,21)
60
61 C      LOAD REM EFF FROM LAST TIME INCR FROM MAT CC INTO MAT DD COL 20
62 C      CALL LODEFF(I,DD,NROW,NCOL,CC,NROW1,NCOL1,LIN2)
63
64 C      FIND CAV PRED FOR THESE REMOVAL EFFICIENCIES
65 C      CALL MASBAL(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
66 C      +      CAVPRD,CDI,NROW2,NCOL2,CFINAL,CEQLIB,LIN,LIN2)
67 C
68 C      SET CAV IN PRED DD(I,22)=CAV IN CALC DD(I,17)
69 C      DO 100 J=1,LIN2
70 C      DD(J,22)=DD(J,17)
71 C      100 CONTINUE
72 C
73 C      RETURN
74 C      ***** END OF SUBROUTINE PREDCT *****
75 C      END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C      *****
2 C      *              SUBROUTINE RAFILL              *
3 C      *  SUBROUTINE TO FILL ADJUSTABLE SIZE REAL ARRAY WITH ZEROS  *
4 C      *****
5      SUBROUTINE RAFILL (XX,NROW,NCOL)
6      INTEGER NROW,NCOL
7      REAL XX(NROW,NCOL)
8
9 C      XX=ARRAY NAME
10 C     NCOL= COLUMNS IN MATRIX
11 C     NROW= ROWS IN MATRIX
12
13         DO 110 I=1,NROW
14         DO 100 J=1,NCOL
15             XX(I,J)=0.0
16 100 CONTINUE
17 110 CONTINUE
18     RETURN
19 C     *****  END OF SUBROUTINE RAFILL  *****
20     END

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0

NUMBER OF ERRORS IN PROGRAM UNIT: 0

NUMBER OF WARNINGS IN COMPILATION : 0

NUMBER OF ERRORS IN COMPILATION : 0

```

1 C      *****
2 C      * FILE:RCHBD.FOR
3 C      * SUBROUTINE FOR REM EFF-RADIAL FLOW CHARCOAL BED
4 C      * DOESNT ALLOW FOR DESORPTION
5 C      * ASSUMES RELATIVELY THIN BED (OD CLOSE TO ID)
6 C      *****
7 C
8 C      SUBROUTINE RCHBD (TN,TN1,CIIN,TCABIN,COEXIS,BEDQ,EMAX,CARTL,
9 C      +BEDOD,BEDID,DENCH,TRTTYP,DCONT,VMOL,MW,VCONC,SOL,SMR,EFF)
10 C      OUTPUT:
11 C          EFF=BED REMOVAL EFF(DEC)
12 C      INPUTS:
13 C          TN,TN1=INCREMENT INITIAL AND FINAL TIMES(HR)
14 C          CIIN=BED INLET CONT CONC (MG/CU M)
15 C          TCABIN=CABIN TEMP (DEG K)
16 C          COEXIS=COEXISTANCE FACTOR
17 C          BEDQ=BED FLOW RATE(CU M/HR)
18 C          EMAX=MAXIMUM BED EFF (DEC)
19 C          CARTL=CARTRIDGE LENGTH (M)
20 C          BEDOD=BED OUTSIDE DIAMETER (M)
21 C          BEDID=BED INSIDE DIAMETER (M)
22 C          DENCH=DENSITY OF CHARCOAL IN BED (KG/CU M)
23 C          TRTTYP=BED TREATMENT TYPE(1=CI CHAR,2=PHOS ACID, OTHER #=NONE)
24 C          DCONT=CONT LIQUID DENSITY (GM/CC)
25 C          VMOL=CONT MOLAR VOL(GM/CC)
26 C          MW=CONT MOLECULAR WGT
27 C          VCONC=CONT VAPOR CONCENTRATION AT TCABIN (MG/CU M)
28 C          SOL=CONTAMINANT HENRY'S LAW CONSTANT (ATM/MOL FRACTION)
29 C          SMR=SUM OF CONT MASS STORED IN BED(MG)-FROM LAST INCR
30 C
31 C      REAL LPREV, LAVN1, LUTIL, LIMM, LAVAV, LADS,MW
32 C      INTEGER FACID,FCI
33 C
34 C      SET CIN=CIIN (THIS PREVENTS CIN FROM BEING PASSED BACK UP
35 C      TO OTHER SUBROUTINES IF IT IS SET TO 1E-20)
36 C      CIN=CIIN
37 C
38 C      BED TREATMENT LOGIC
39 C          FACID=FLAG IF BED IS TREATED WITH PHOSPHORIC ACID (Y=1 N=0)
40 C          FCI=FLAG FOR CI CHAR IN BED (REMOVES FORMALDAHYDE)
41 C      IF (NINT(TRTTYP).EQ.2) THEN
42 C          FACID=1
43 C          FCI=0
44 C      ELSEIF (NINT(TRTTYP).EQ.1) THEN
45 C          FACID=0
46 C          FCI=1
47 C      ELSE

```

```

48         FACID=0
49         FCI=0
50     ENDIF
51 C
52 C     TEST FOR NO BED FLOW(BEDQ=<0) OR TN-TN1<=0;BEDL,BEDDIA,DENCH=0
53     IF((BEDQ.LE.0).OR.(TN-TN1.LE.0).OR.(CARTL.LE.0).OR.(BEDOD.LE.0)
54 +.OR.(DENCH.LE.0)) THEN
55         EFF=0
56         GOTO 199
57     ENDIF
58 C     TEST FOR CI CHARCOAL AND FORMALDEHYDE(FCI=1 AND MW=30.03)
59     IF((MW.EQ.30.03).AND.(FCI.EQ.1)) THEN
60         CALL RCICH(EFF,EMAX,CARTL,BEDOD,BEDID,DENCH,SMR,BEDQ)
61         GOTO 199
62     ENDIF
63 C
64 C     TEST FOR AMMONIA AND PHOS ACID ON CHAR(FACID=1 AND MW=17.0 )
65     IF ((MW.EQ.17.0).AND.(FACID.EQ.1)) THEN
66         CALL RACCH(EFF,EMAX,CARTL,BEDOD,BEDID,DENCH,SMR)
67         GOTO 199
68     ENDIF
69 C
70 C     TEST FOR MOL VOL=0 (NO CHAR REMOVAL)
71     IF (VMOL.EQ.0) THEN
72         EFF=0
73         GOTO 199
74     ENDIF
75 C
76 C     CHARCOAL REMOVAL EFFICIENCY CALCULATION
77 C     BED LENGTH (M)-ASSUMES THIN BED
78     BEDL=(BEDOD-BEDID)/2
79     IF (BEDL.LT.0) BEDL=0
80 C     BED WGT (KG)
81     BEDWGT=DENCH*.785*(BEDOD**2-BEDID**2)*CARTL
82 C     SUPERFICIAL BED VEL(FT/MIN)
83     BEDVEL=BEDQ*.0348/((BEDOD+BEDID)*CARTL)
84 C     TEST FOR CIN TOO SMALL IN AVAL CALC
85     IF (CIN.LT.1E-20) CIN=1E-20
86     AVAL=(TCABIN/VMOL)*LOG10(VCONC/CIN)
87 C     ADS ZONE LENGTH FOR 90% REMOVAL (M)
88     LADS=AVAL*.000275*(BEDVEL/1.3)**.8
89 C     GET QI(CC LIQ CONT/GM CHAR)
90     CALL FQI(AVAL,QI,FACID,SOL)
91 C     LENGTH OF BED PREVIOUSLY USED BY CONT AT THIS C INLET (M)
92     LPREV=SMR*1.0E-6*COEXIS*BEDL/(DCONT*BEDWGT*QI)
93 C     RATE OF BED USAGE (M BED/ MG CONT)
94     LIMM=1.0E-6*COEXIS*BEDL/(DCONT*BEDWGT*QI)
95 C     LENGTH OF BED AVAILABLE FOR ADS ZONE AT BEGINNING OF INCR (M)
96     LAVN1=BEDL-LPREV
97     IF (LAVN1.LT.0) LAVN1=0

```



```

98 C      FIX HERE IF DESORPTION IS DESIRED
99      IF (LAVN1/LADS.GT.20) THEN
100      EFFAV=EMAX
101      ELSE
102 C      INIT INCR EFF BASED ON C IN AND BED L AVAIL AT BEG OF INCR(DEC)
103      EFAVN1=EMAX*(1-EXP(-2.3025851*LAVN1/LADS))
104 C      LOOP FOR EFFICIENCY
105      EFFAV=EFAVN1
106      DO 399 J=1,10,1
107 C      LENGTH OF BED UTILIZ IN INCR (M)
108      LUTIL=CIN*BEDQ*EFFAV*(TN-TN1)*LIMM
109      IF (LUTIL.GT.LAVN1) THEN
110      GOTO 299
111      ELSE
112 C      AVERAGE BED LENGTH AVAIL (M)
113      LAVAV=LAVN1-LUTIL/2
114      IF ((LAVAV/LADS).GE.20) THEN
115      EFFAV=EMAX
116      GOTO 299
117      ELSE
118 C      AV EFF BASED ON AV BED L AVAIL (DEC)
119      EFFAV=EMAX*(1-EXP(-2.3025851*LAVAV/LADS))
120      ENDIF
121      ENDIF
122      399 CONTINUE
123      299 ENDIF
124 C      MAX EFF BASED ON C IN AND RATE OF BED USAGE (DEC)
125      EFFMAX=LAVN1/(CIN*BEDQ*(TN-TN1)*LIMM)
126      IF (EFFAV.GT.EFFMAX) EFFAV=EFFMAX
127      IF (EFFAV.LT.0) EFFAV=0
128      IF (EFFAV.GT.EMAX) EFFAV=EMAX
129 C      EFF=ACTUAL EFF OUTPUT FROM SUBROUTINE
130      EFF=EFFAV
131 C      REMOVE THIS CHECK IF DESORPTION IS ADDED
132      199 IF (EFF.LT.0) EFF=0
133      IF (EFF.GT.EMAX) EFF=EMAX
134      RETURN
135      END

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS IN PROGRAM UNIT: 0

```

136 C      *****
137 C      * SUBROUTINE RACCH - CALCULATES REMOVAL EFF *
138 C      * BED WITH NH3 AND .2 MILLIMOLE H3PO4 ON CHAR *
139 C      *****
140      SUBROUTINE RACCH(EFF,EMAX,CARTL,BEDOD,BEDID,DENCH,SMR)
141 C      OUTPUTS
142 C      EFF=OUTPUT REMOVAL EFF (DEC)
143 C      INPUTS

```

```

144 C      EMAX=MAXIMUM BED REMOVAL EFF (DEC)
145 C      CARTL=CARTRIDGE LENGTH (M)
146 C      BEDOD=BED OUTSIDE DIAMETER (M)
147 C      BEDID=BED INSIDE DIAMETER (M)
148 C      DENCH=CHARCOAL DENSITY (KG/CU M)
149 C      SMR=SUM OF MASS OF CONT REMOVED AT BEG OF INCR (MG)
150 C
151 C      CHAR USED (KG)
152 C      CHRUSD=9.61E-6*SMR
153 C      CHAR BED WGT (KG)
154 C      BEDWGT=CARTL*(BEDOD**2-BEDID**2)*.785*DENCH
155 C      IF (CHRUSD.LT.0.8*BEDWGT) THEN
156 C          EFF=EMAX
157 C      ELSE
158 C          EFF=EMAX*SIN((BEDWGT-CHRUSD)*1.57/(BEDWGT*0.2))
159 C      ENDIF
160 C      PREVENTS NEGATIVE EFF FOR REACTION
161 C      IF (EFF.LT.0) EFF=0
162 C      IF (EFF.GT.EMAX) EFF=EMAX
163 C      RETURN
164 C      END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

```

```

165 C      *****
166 C      * SUBROUTINE RCICH - CALCULATES REMOVAL EFF *
167 C      * FOR FORMALDELYDE AND CI CHAR BED *
168 C      *****
169 C      SUBROUTINE RCICH(EFF,EMAX,CARTL,BEDOD,BEDID,DENCH,SMR,BEDQ)
170 C      OUTPUTS
171 C          EFF=OUTPUT REMOVAL EFF (DEC)
172 C      INPUTS
173 C          EMAX=MAXIMUM BED REMOVAL EFF (DEC)
174 C          CARTL=CARTRIDGE LENGTH (M)
175 C          BEDOD=BED OUTSIDE DIAMETER (M)
176 C          BEDID=BED INSIDE DIAMETER (M)
177 C          DENCH=CHARCOAL DENSITY (KG/CU M)
178 C          SMR=SUM OF MASS OF CONT REMOVED AT BEG OF INCR (MG)
179 C          BEDQ=BED FLOW RATE (CU M/HR)
180 C
181 C          BEDWGT=CARTL*(BEDOD**2-BEDID**2)*.785*DENCH
182 C          PERCENT OF BED WEIGHT CONSUMED (DEC)
183 C          PBWGT=SMR/(BEDWGT*1E6)
184 C          IF (PBWGT.LT..0012) THEN
185 C              EFF=1-PBWGT*83.3
186 C          ELSE
187 C              EFF=.9*COS(PBWGT*1.57/.05)
188 C          ENDIF
189 C      BED RESIDENCE TIME (SEC)

```

```

190      BREST=(BEDOD-BEDID)*CARTL*(BEDOD+BEDID)*3600/(BEDQ*1.273)
191      IF(BREST.LT.0.25)THEN
192          EFF=EFF*BREST/.25
193      ENDIF
194 C      PREVENTS NEGATIVE EFF FOR REACTION
195 C      IF (EFF.LT.0) EFF=0
196      IF(EFF.GT.EMAX) EFF=EMAX
197      RETURN
198      END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C      *****
2 C      *                REGENERATION SUBROUTINE-REGEN                *
3 C      * FOR ALL BEDS (3-15) DETERMINES IF BED IS TO BE REGENERATED *
4 C      * AT BEGINNING OF TIME INCREMENT, AND IF REGENERATION IS TO *
5 C      * DURING THE ENTIRE TIME INCREMENT- IF THE BED IS TO BE *
6 C      * REGENERATED THE MASSES STORED ARE SET TO ZERO, AND IF *
7 C      * REGENERATION IS TO OCCUR THROUGHOUT THE TIME INCREMENT THE *
8 C      * BED FLOW RATE IS SET TO ZERO; OTHERWISE IT IS SET TO THE *
9 C      * ORIGINAL VALUE *
10 C      *****
11      SUBROUTINE REGEN(TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
12      +CDI,NROW2,NCOL2,LIN,LIN2,IMSGDN)
13      INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2
14      REAL DD(NROW,NCOL)
15      REAL CC(NROW1,NCOL1)
16      REAL CDI(NROW2,NCOL2)
17
18 C NOTE: BEFORE RUNNING THIS SUBROUTINE THE ORIGINAL FLOW RATES FROM
19 C      MAT DD COL 2 MUST BE SAVED IN MAT DD COL 7- AT THE END OF THE
20 C      TIME INCREMENT THEY MUST BE RESTORED TO COL 2
21 C DIRECT INPUTS:
22 C      TN=INCREMENT END TIME(HRS); TN1=INCR BEGINNING TIME HRS
23 C      DD,NROW,NCOL=NAME AND SIZE OF MAT DD
24 C      CC,NROW1,NCOL1=NAME AND SIZE OF MAT CC
25 C      CDI,NROW2,NCOL2=NAME AND SIZE OF MAT CDI
26 C      LIN=NO. OF CONT IN MAT CDI
27 C      LIN2=NO. OF DEVICES IN MAT DD
28
29 C OTHER INPUTS FROM MAT DD
30 C      TIR=INITIAL (FIRST) REGENERATION TIME (HRS)
31 C      TRCI=REGEN/CHANGEOUT INTERVAL (HRS)
32 C      TRD=REGENERATION DURATION (HRS)
33 C      DEVICE NO., TYPE, FLOW RATE, ETC
34
35 C OUTPUTS:
36 C      A) IF REGENERATION OCCURS AT THE BEGINNING OF ANY TIME INCREMENT
37 C          1) FOR ANY DEVICE WHICH IS A CHARCOAL BED
38 C              FOR ALL CONT 1 TO LIN IT PUTS SUM MASS REM=0 IN MAT CC
39 C              COL 12,15,18....48 AS APPROPRIATE FOR THAT DEVICE
40 C          2) FOR ANY DEVICE WHICH IS A LIOH BED
41 C              IT DOES 1) ABOVE, AND IN ADDITION PUTS SUM MASS REM=0 IN
42 C              MAT DD COL 16 FOR THAT DEVICE
43 C      B) IF REGENERATION IS OCCURRING THROUGHOUT THE WHOLE INTERVAL
44 C          IT SETS Q OF DEVICE=0; IF REGENERATION IS NOT OCCURRING, IT
45 C          SETS Q=THE ORIGINAL VALUE
46 C
47 C      SUBROUTINES REQUIRED:

```

```

48 C      REGCHG
49 C
50 C      START LOOP FOR ALL DEVICES 3 TO 15
51      DO 100 J=3,LIN2
52 C      IF DEVICE DOES NOT EQUAL CHARCOAL OR LIOH THEN GO TO END OF LOOP
53      IF (DD(J,3).NE.3.AND.DD(J,3).NE.4.AND.DD(J,3).NE.5) GOTO 100
54
55 C      ASSIGN PROPER VARIABLES FOR DEVICE
56 C      DEVICE = CHARCOAL
57      IF (DD(J,3).EQ.3.OR.DD(J,3).EQ.4) THEN
58 C      REGENERATION/CHANGEOUT INTERVAL (HRS)
59      TRCI=DD(J,15)
60 C      REGENERATION DURATION (HRS)
61      TRD=DD(J,16)
62 C      INITIAL(FIRST) REGENERATION
63      TIR=DD(J,14)
64      ENDIF
65
66 C      DEVICE = LIOH
67      IF (DD(J,3).EQ.5) THEN
68 C      REGENERATION/CHANGEOUT INTERVAL (HRS)
69      TRCI=DD(J,14)
70 C      REGENERATION DURATION (HRS)
71      TRD=0
72 C      INITIAL(FIRST) REGENERATION
73      TIR=DD(J,13)
74      ENDIF
75
76 C      CHECK AND FIX INPUT AS REQ + PRINT WARNINGS
77 C      TIME INCREMENT (HRS)
78      TINC=DD(1,11)
79 C      INITIAL TIME NOT EQUAL TO MULTIPLE OF TIME INCREMENT
80      IF (AINT(TIR/TINC).NE.(TIR/TINC)) THEN
81      TIR=AINT(TIR/TINC)*TINC
82      OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
83      WRITE(IMSGDN,*) 'INCREMENT BEGINNING TIME ',TN1,
84      + 'DEV NO.',(J)
85      WRITE(IMSGDN,*) 'INITIAL TIME NOT = MULTIPLE OF TIME
86      + INCREMENT'
87      WRITE(IMSGDN,*) 'TRUNCATED TO ',TIR
88      CLOSE(IMSGDN)
89      ENDIF
90
91 C      REGEN/CHGOUT INTERVAL < TIME INCR OR NOT= MULTIPLE OF TIME INCR
92      IF (AINT(TRCI/TINC).NE.(TRCI/TINC)) THEN
93      TRCI=AINT(TRCI/TINC)*TINC
94      OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
95      WRITE(IMSGDN,*) 'INCREMENT BEGINNING TIME ',TN1,
96      + 'DEV NO.',(J)
97      WRITE(IMSGDN,*) 'REGEN/CHG TIME NOT = MULTIPLE OF TIME

```

```

98      + INCREMENT'
99          WRITE(IMSGDN,*) 'TRUNCATED TO ',TRCI
100         CLOSE(IMSGDN)
101     ENDIF
102
103 C      IF REGEN/CHGOUT INTERVAL <= 0 THEN GOTO END OF LOOP FOR CONT
104         IF (TRCI.LE.0) THEN
105             GOTO 100
106         ENDIF
107
108 C      REGEN DURATION < OR NOT = MULTIPLE OF TIME INCREMENT
109         IF (AINT(TRD/TINC).NE.(TRD/TINC)) THEN
110             TRD=AINT(TRD/TINC)*TINC
111             OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
112             WRITE(IMSGDN,*) 'INCREMENT BEGINNING TIME ',TN1,
113 + 'DEV NO.',(J)
114             WRITE(IMSGDN,*) 'REG DURATION NOT = MULTIPLE OF TIME
115 + INCREMENT'
116             WRITE(IMSGDN,*) 'TRUNCATED TO ',TRD
117             CLOSE(IMSGDN)
118         ENDIF
119
120 C      REGENERATION DURATION > REGEN/CHG INTERVAL
121         IF (TRD.GT.TRCI) THEN
122             TRD=TRCI
123             OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
124             WRITE(IMSGDN,*) 'INCREMENT BEGINNING TIME ',TN1,
125 + 'DEV NO.',(J)
126             WRITE(IMSGDN,*) 'REGEN DURATION > REGEN/CHG INTERVAL'
127             WRITE(IMSGDN,*) 'TRUNCATED TO ',TRD
128             CLOSE(IMSGDN)
129         ENDIF
130
131 C      CHECK TO SEE IF REGENERATION OCCURS AT BEGINNING OF TIME INCR,
132 C      AND IF REGEN OCCURS THROUGHOUT WHOLE TIME INCREMENT
133
134      CALL REGCHG(TN1,TRCI,TRD,TIR,TINC,IRBFLG,IRTF LG)
135 C      REGENERATION OCCURS AT BEGINNING OF INCREMENT
136         IF (IRBFLG.EQ.1) THEN
137 C          PUT SUM MASS REM =0 IN MAT CC FOR THIS DEVICE
138 C          START LOOP FOR ALL CONT FOR THIS DEVICE
139             K=J*3+3
140             DO 101 I=1,LIN
141                 CC(I,K)=0
142     101      CONTINUE
143
144 C      IF DEVICE = LIOH BED PUT SUM MASS=0 IN DD(J,16)
145         IF (DD(J,3).EQ.5) THEN
146             DD(J,16)=0
147         ENDIF

```

```

148             ENDIF
149
150 C           REGENERATION OCCURS THROUGHOUT ENTIRE INCREMENT
151             IF (IRTFLG.EQ.1) THEN
152 C           SET DEVICE Q=0
153             DD(J,2)=0
154             ELSE
155 C           SET DEVICE Q= ORIGINAL VALUE
156             DD(J,2)=DD(J,7)
157             ENDIF
158
159 C   END OF J LOOP FOR EACH DEVICE
160   100 CONTINUE
161
162   RETURN
163   END

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS IN PROGRAM UNIT: 0

```

164
165
166 C   *****
167 C   *           AUXILIARY REGENERATION SUBROUTINE-REGCHG           *
168 C   *   DETERMINES IF REGEN/CHANGEOUT IS TO OCCUR AT BEGINNING OF   *
169 C   *   TIME INCREMENT-ALSO DETERMINES IF REGENERATION IS OCCURRING *
170 C   *   THROUGHOUT THE TIME INCREMENT                               *
171 C   *****
172   SUBROUTINE REGCHG(TN1,TRCI,TRD,TIR,TINC,IRBFLG,IRTFLG)
173 C
174 C   INPUTS:
175 C   TN1=INCREMENT INITIAL TIME (HRS)
176 C   TRCI=CHANGEOUT/REGENERATION INTERVAL (HRS)
177 C   TRD=REGENERATION DURATION (HRS)
178 C   TIR=INITIAL (FIRST) REGENERATION TIME (HRS)
179 C   TINC=TIME INCREMENT (HRS)
180 C
181 C   OUTPUTS:
182 C   REGENERATION OCCURS AT BEGINNING OF TIME INCREMENT (Y OR N)
183 C   (IRBFLG=1 FOR Y & 0 FOR N)
184 C   REGENERATION IS OCCURRING THROUGHOUT THE WHOLE INCREMENT (Y OR N)
185 C   (IRTFLG=1 FOR Y & 0 FOR N)
186 C
187
188 C   REGENERATION OCCURS AT BEGINNING OF TIME INCREMENT
189
190   IF (TN1.EQ.0) GOTO 10
191   IF(TN1.LT.TIR) GOTO 10
192   IF(TRCI.LE.0) GOTO 10
193   IF(AINT((TN1-TIR)/TRCI).EQ.((TN1-TIR)/TRCI)) THEN

```

```

194 C      REGENERATION OCCURS
195          IRBFLG=1
196          GO TO 20
197          ENDIF
198 C      NO REGENERATION OCCURS
199 010 IRBFLG=0
200 020 CONTINUE
201
202 C      REGENERATION OCCURRING THROUGHOUT ENTIRE TIME INCREMENT
203
204          IF (TRCI.LE.0) GOTO 30
205          IF ((TRD.LE.0).OR.(TN1.LT.TIR)) GOTO 30
206          IF (TN1.GE.AINT((TN1-TIR)/TRCI)*TRCI+TIR+TRD) THEN
207              GOTO 30
208          ELSE
209 C      REGENERATION OCCURS
210          IRTFLG=1
211          GOTO 40
212          ENDIF
213
214 C      REGEN DOESN'T OCCUR
215 030 IRTFLG=0
216
217 040 CONTINUE
218          RETURN
219          END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```



```

1 C *****
2 C * SUBROUTINE RINCDD *
3 C * SUBROUTINE TO OPERATE ON INCREMENT DEPENDENT DATA *
4 C * READS DATA FROM MAT TT AND PUT IT IN THE PROPER PLACES IN *
5 C * MAT CDI OR MAT DD - USED AT THE BEGINNING OF EACH TIME INCR *
6 C *****
7
8 SUBROUTINE RINCDD(I,TN,TN1,DD,NROW,NCOL,LIN2,
9 +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,LIN,TT,NTTROW,NTTCOL,LIN1)
10 INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2,NTTROW,NTTCOL
11 REAL DD(NROW,NCOL)
12 REAL CC(NROW1,NCOL1)
13 REAL CDI(NROW2,NCOL2)
14 REAL TT(NTTROW,NTTCOL)
15
16 C SUBROUTINES REQUIRED:
17 C NONE
18 C DD,NROW,NCOL,LIN2=NAME,DIM & NO DEV IN MAT DD
19 C CC,NROW1,NCOL1=NAME & DIM OF MAT CC
20 C CDI,NROW2,NCOL,LIN=NAME,DIM & NO CONT IN MAT CDI
21 C TT,NTTROW,NTTCOL,LIN1=NAME,DIM & NO ITEMS IN MAT TT
22
23 IF (LIN1.EQ.0) GOTO 999
24 C BEGIN LOOP FOR ALL LINES IN MAT TT
25 DO 100 K=1,LIN1
26
27 C CHECK FOR TIME >= TN1 AND < TN
28 IF((TT(K,1).LT.TN1).OR.(TT(K,1).GE.TN)) THEN
29 GO TO 100
30 ENDIF
31
32 C IDENTIFY VARIABLES
33 ICONTN=NINT(TT(K,2))
34 GENRT=TT(K,3)
35 IDEVNO=NINT(TT(K,4))
36 DEVQ=TT(K,5)
37 ICOLNO=TT(K,6)
38 VAL=TT(K,7)
39
40 C ICONTN=INTEGER CONTAMINANT NO. - TT(K,2)
41 C GENRT=CONT GENERATION RATE (MG/HR) - TT(K,3)
42 C IDEVNO=INTEGER DEVICE NUMBER - TT(K,4)
43 C DEVQ=DEVICE FLOW RATE (CU M/HR) - TT(K,5)
44 C ICOLNO=INTEGER COLUMN NUMBER IN MAT DD - TT(K,6)
45 C VAL=NEW VALUE IN MAT DD - TT(K,7)
46
47 C CASE NO. 1 - CHANGE CONTAMINANT GENERATION RATE

```

```

48
49      IF((ICONTN.GT.0).AND.(ICONTN.LE.LIN)) THEN
50          IF(IDEVNO.EQ.1) THEN
51              CDI(ICONTN,1)=GENRT
52          ELSEIF ((IDEVNO.GE.3).AND.(IDEVNO.LE.LIN2)) THEN
53              CDI(ICONTN,(7+IDEVNO))=GENRT
54          ENDIF
55          GOTO 100
56      ENDIF
57
58 C      CASE 2 - CHANGE DEVICE FLOW OR OTHER DD DATA
59 C      THIS CASE WORKS ONLY IF ANY CONT NO. <=0
60 C      MUST USE -1 FOR ANY Q OR NEW VALUE NOT TO BE CHANGED
61 C      MAT DD COL NO. <0 ALSO STOPS NEW VALUE FROM BEING CHANGED
62
63      IF(ICONTN.LE.0) THEN
64          IF((IDEVNO.GE.1).AND.(IDEVNO.LE.LIN2)) THEN
65              IF(DEVQ.GE.0) THEN
66 C              CHANGE DEVICE FLOW IN MAT DD
67                  DD(IDEVNO,2)=DEVQ
68              ENDIF
69          IF((ICOLNO.GE.1).AND.(ICOLNO.LE.16)) THEN
70              IF (VAL.GE.0) THEN
71 C              CHANGE VALUE IN MAT DD
72                  DD(IDEVNO,ICOLNO)=VAL
73              ENDIF
74          ENDIF
75      ENDIF
76  ENDIF
77      100 CONTINUE
78      999 RETURN
79      END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C      *****
2 C      *          SUBROUTINE  RRIN          *
3 C      *      SUBROUTINE TO READ REAL DATA INTO MAT XX(ROW,COL)      *
4 C      *      RETURNS NUMBER OF LINES OF DATA READ FROM FILE        *
5 C      *      READS FROM COL 1 TO COL LSTCOL                          *
6 C      *****
7 C      NOTE: INPUT NUMBERS MUST BE SEPARATED BY BLANKS
8          SUBROUTINE RRIN(XX,NROW,NCOL,LSTCOL,LIN)
9          INTEGER NROW,NCOL,IOVAL,LSTCOL,LIN
10         CHARACTER FNAME*24
11         REAL XX(NROW,NCOL)
12         IF(LSTCOL.GT.NCOL) LSTCOL=NCOL
13     010 READ(*,'(A)') FNAME
14         OPEN(1,FILE=FNAME,STATUS='OLD',IOSTAT=IOVAL)
15         IF(IOVAL.NE.0) GOTO 900
16         LIN=0
17         DO 100 I=1,NROW
18             READ(1,*,IOSTAT=IOVAL,END=500,ERR=900 ) (XX(I,J),J=1,LSTCOL)
19             LIN=LIN+1
20     100 CONTINUE
21     500 WRITE(*,'(A)') ' DONE WITH FILE INPUT'
22         WRITE (*,*) ' '
23         CLOSE (1)
24         GOTO 990
25     900 WRITE(*,*) 'IOERROR= ',IOVAL
26         CLOSE (1)
27         WRITE(*,*) 'WHAT IS THE INPUT FILE NAME? '
28         GOTO 10
29     990 RETURN
30 C      ***** END OF SUBROUTINE RRIN *****
31         END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C      *****
2 C      *          SUBROUTINE  RROUT                      *
3 C      *    SUBROUTINE TO WRITE DATA TO CONSOLE, OR PRINTER      *
4 C      *    WRITES REAL DATA FROM MAT XX(ROW,COL)                *
5 C      *    WRITES FROM FSTCOL TO LSTCOL                          *
6 C      *****
7      SUBROUTINE RROUT(XX,NROW,NCOL,FSTCOL,LSTCOL,LIN,IMSGDN,FNAME,
8      + IDEVNO,IOVAL)
9      INTEGER NROW,NCOL,IOVAL,FSTCOL,LSTCOL,LIN,IDEVNO
10     CHARACTER FNAME*24,DES*1
11     REAL XX(NROW,NCOL)
12     IF (FSTCOL.GT.NCOL) FSTCOL=NCOL
13     IF (LSTCOL.GT.NCOL) LSTCOL=NCOL
14     IF (FSTCOL.GT.LSTCOL) FSTCOL=LSTCOL
15 C 010 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
16 C     WRITE(IMSGDN,'(A)') ' WRITE TO LPT1 OR CON OR END '
17 C     CLOSE(IMSGDN)
18 C     READ(*,'(A)') FNAME
19 C     QUIT IF FNAME=END
20 C     IF(FNAME.EQ.'END') GOTO 990
21 C     IF((FNAME.NE.'LPT1').AND.(FNAME.NE.'CON')) GOTO 10
22 C     OPEN(1,FILE=FNAME,IOSTAT=IOVAL)
23     IF(IOVAL.NE.0) GOTO 900
24     DO 110 I=1,LIN
25 C         WRITE(1,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
26         WRITE(IDEVNO,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
27 070     FORMAT(1X,7G11.4)
28 C         WRITE(1,*,IOSTAT=IOVAL,ERR=900)
29         WRITE(IDEVNO,*,IOSTAT=IOVAL,ERR=900)
30 110 CONTINUE
31 C     CLOSE (1)
32     GOTO 990
33 900 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
34     WRITE(IMSGDN,*) 'IOERROR= ',IOVAL
35     CLOSE(IMSGDN)
36 C     CLOSE (1)
37     CLOSE (IDEVNO)
38 C     GOTO 10
39 990 RETURN
40 C     *****      END OF SUBROUTINE RROUT      *****
41     END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C      *****
2 C      *          SUBROUTINE  RROUT2                      *
3 C      *          SUBROUTINE TO WRITE DATA TO CONSOLE, OR PRINTER      *
4 C      *          WRITES REAL DATA FROM MAT XX(ROW,COL)                *
5 C      *          WRITES FROM FSTCOL TO LSTCOL                          *
6 C      *****
7      SUBROUTINE RROUT2 (XX,NROW,NCOL,FSTCOL,LSTCOL,LIN,IMSGDN)
8      INTEGER NROW,NCOL,IOVAL,FSTCOL,LSTCOL,LIN,IDEVNO
9      CHARACTER FNAME*24,DES*1
10     REAL XX(NROW,NCOL)
11     IF (FSTCOL.GT.NCOL) FSTCOL=NCOL
12     IF (LSTCOL.GT.NCOL) LSTCOL=NCOL
13     IF (FSTCOL.GT.LSTCOL) FSTCOL=LSTCOL
14 010 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
15     WRITE(IMSGDN,'(A)') ' WRITE TO LPT1 OR CON OR END '
16     CLOSE(IMSGDN)
17     READ(*,'(A)') FNAME
18     QUIT IF FNAME=END
19     IF(FNAME.EQ.'END') GOTO 990
20     IF((FNAME.NE.'LPT1').AND.(FNAME.NE.'CON')) GOTO 10
21     OPEN(1,FILE=FNAME,IOSTAT=IOVAL)
22     IF(IOVAL.NE.0) GOTO 900
23     DO 110 I=1,LIN
24         WRITE(1,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
25 070   FORMAT(1X,7G11.4)
26         WRITE(1,*,IOSTAT=IOVAL,ERR=900)
27 110 CONTINUE
28     CLOSE (1)
29     GOTO 990
30 900 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
31     WRITE(IMSGDN,*) 'IOERROR= ',IOVAL
32     CLOSE(IMSGDN)
33     CLOSE (1)
34     GOTO 10
35 990 RETURN
36 C      *****  END OF SUBROUTINE RROUT *****
37     END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```

```

1 C      *****
2 C      *                      SUBROUTINE-SLIOH                      *
3 C      * SUM LIOH USED IN TIME INCREMENT FOR EACH BED ONE AT A TIME *
4 C      * AND FOR ALL CONTAMINANTS FOR EACH BED                       *
5 C      *****
6
7      SUBROUTINE SLIOH(TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
8 +CDI,NROW2,NCOL2,LIN,LIN2)
9      INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2
10     REAL DD(NROW,NCOL)
11     REAL CC(NROW1,NCOL1)
12     REAL CDI(NROW2,NCOL2)
13 C
14 C SUBROUTINES REQUIRED: NONE
15 C DIRECT INPUTS:
16 C   TN=INCREMENT END TIME(HRS); TN1=INCR BEGINNING TIME HRS
17 C   DD,NROW,NCOL=NAME AND SIZE OF MAT DD
18 C   CC,NROW1,NCOL1=NAME AND SIZE OF MAT CC
19 C   CDI,NROW2,NCOL2=NAME AND SIZE OF MAT CDI
20 C   LIN=NO. OF CONT IN MAT CDI
21 C   LIN2=NO. OF DEVICES IN MAT DD
22
23 C OTHER INPUTS FROM MAT DD
24 C   DD(J,3)=DEVICE NUMBER
25 C   DD(J,16)=AMT OF LIOH PREVIOUSLY USED BY DEVICE
26 C   CDI(I,17)=LB LIOH UTIL/LB CONT ADSORBED IN BED (FOR ONE CONT)
27
28 C OUTPUTS (STORED IN MAT DD):
29 C   DD(J,16)=AMOUNT OF LIOH UTILIZED BY DEVICE THROUGH THE END OF
30 C   THIS TIME INCREMENT
31 C   DD(J,15)=RATE OF LIOH USAGE FOR DEVICE
32 C
33     K=11
34
35 C   START LOOP FOR ALL DEVICES 3 TO 15
36     DO 100 J=3,LIN2
37 C     CHECK FOR DEVICE = LIOH BED
38     IF (DD(J,3).EQ.5) THEN
39 C     RATE OF LIOH UTILIZATION (KG/HR)
40     RWUTLI=0
41 C     BEGIN LOOP FOR ALL CONTAMINANTS
42     DO 110 I=1,LIN
43     RWUTLI=RWUTLI+CC(I,K)*CDI(I,7)*1E-6
44 110 CONTINUE
45 C     STORE RATE OF LIOH UTILIZATION IN MAT DD FOR THIS DEVICE
46     DD(J,15)=RWUTLI
47 C     UPDATE AMOUNT OF LIOH UTIL THROUGH THE END OF TIME INCR(KG)

```

```

48          DD(J,16)=DD(J,16)+RWUTLI*(TN-TN1)
49          ENDIF
50          K=K+3
51 C        END J LOOP
52 100      CONTINUE
53          RETURN
54          END

```

```

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0

```


APPENDIX B
TOXIC HAZARD INDEX DESCRIPTION

The toxic hazard index, or T-value, is the method used by toxicologists to assess the acceptability of an atmosphere containing a mixture of contaminants. This approach is derived from the American Conference of Governmental Industrial Hygienists guidelines for setting threshold limit values for contaminant mixtures. Since the effects on humans of many atmospheric contaminants are considered to be additive this mixture approach is applied to sixteen contaminant groups. The groups considered in the T-value calculation used in the TCCS Computer Program are the following:

1. Alcohols
2. Aldehydes
3. Aromatic hydrocarbons
4. Esters
5. Ethers
6. Chlorocarbons
7. Chlorofluorocarbons
8. Fluorocarbons
9. Hydrocarbons
10. Inorganic acids
11. Ketones
12. Mercaptans and sulfides
13. Nitrogen oxides
14. Organic acids
15. Organic nitrogens
16. Miscellaneous

The group numbers used in the computer program output correspond to the above group listing.

The T-value is calculated for each group by calculating the sum of the ratios of the contaminants' concentrations to their maximum allowable concentration while the overall T-value is the sum of the group T-values for the alcohols, aldehydes, aromatic hydrocarbons, esters, ethers, hydrocarbons, inorganic acids, ketones, nitrogen oxides, organic acids, and miscellaneous groups. These calculations are conducted according to the following equations:

$$T_{\text{group}} = \sum C_c / C_m \quad (\text{B1})$$

$$T_{\text{overall}} = \sum T_{\text{group}} , \quad (\text{B2})$$

where C_c is the contaminant concentration in the atmosphere in mg/m^3 and C_m is the maximum allowable concentration in the atmosphere in mg/m^3 .

The criteria for acceptability are the following:

1. The T-value for each group must be less than one
2. The overall T-value must be less than one

If either of these criteria are exceeded, the atmosphere is considered unacceptable.

REFERENCES

1. "Design and Fabrication of a Trace Contaminant Removal System for Apollo," Phase I Report, NASA Contract NAS9-3415. Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, March 15, 1965.
2. Ibid., Phase II Report. November 23, 1965.
3. Ibid., Phase III Report. April 14, 1966.
4. Olcott, T.M.: "Study and Preliminary Design of an Isotope-Heated Catalytic Oxidizer System." NASA CR-66346, NASA Contract NAS 1-6256. Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, 1969.
5. Robell, A.J., Arnold, C.R., Wheeler, A., Kersels, G.J., and Merrill, R.P.: "Trace Contaminant Adsorption and Sorbent Regeneration." NASA CR-1582. Lockheed Missiles and Space Co., Palo Alto, CA, September 1970.
6. Ibid., pp. 32-70.
7. Olcott, T.M.: "Development of a Sorber Trace Contaminant Control System Including Pre- and Post-Sorbers for a Catalytic Oxidizer." NASA CR-2027. Lockheed Missiles and Space Co., Sunnyvale, CA, May 1972, pp. D1-D18.
8. Olcott, T.M.: "Design, Fabrication, and Test of a Trace Contaminant Control System." LMSC-D462467. Biotechnology Organization Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, November 28, 1975.
9. Jagow, R.B., Jaffe, R.J., and Lamparter, R.A.: "Development of a Computer Program for Spacelab Contaminant Control Analysis." LMSC-D556710, NASA Contract NAS 8-31551. Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, January 15, 1977.
10. Lamparter, R.A., and Yee, R.Q.: "A Generalized Computer Program for Trace Contaminant Control." LMSC-D564045, NASA Contract NAS 8-32400. Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, September 1, 1977.
11. "Preliminary Documentation for Space Station Gaseous Contaminant Control Program." NASA Contract NAS8-36406. Bioastronautics Division, Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, September 5, 1986.
12. "Flammability, Odor, and Offgassing Requirements and Test Procedures for Material in Environments that Support Combustion." NASA Handbook NHB 8060.1B. National Aeronautics and Space Administration Office of Space Transportation Systems, Washington, DC, September 1981, pp. D10-D11.
13. "Preliminary Documentation for Space Station Gaseous Contaminant Control Program." NASA Contract NAS8-36406. Bioastronautics Division, Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, September 5, 1986, pp. 33-61.

14. Olcott, T.M.: "Development of a Sorber Trace Contaminant Control System Including Pre- and Post-Sorbers for a Catalytic Oxidizer." NASA CR-2027, Lockheed Missiles and Space Co., Sunnyvale, CA, May 1972, p. 26.
15. Ibid., p. 24
16. Robell, A.J., Arnold, C.R., Wheeler, A., Kersels, G.J., and Merrill, R.P.: "Trace Contaminant Adsorption and Sorbent Regeneration." NASA CR-1582. Lockheed Missiles and Space Co., Palo Alto, CA, September 1970.
17. Olcott, T.M.: "Development of a Sorber Trace Contaminant Control System Including Pre- and Post-Sorbers for a Catalytic Oxidizer." NASA CR-2027, Lockheed Missiles and Space Co., Sunnyvale, CA, May 1972.
18. Ibid., p. 31.
19. "Preliminary Documentation for Space Station Gaseous Contaminant Control Program." NASA Contract NAS8-36406, Bioastronautics Division, Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, September 5, 1986, p. 59.
20. Jagow, R.B., Katan, T., Lamparter, R.A., Dibble, K.P., Masi, O.H., Richardi, C.L., and Kawasaki, E.H.: "Investigation and Testing of Low Temperature Carbon Monoxide Catalysts." LMSC-D556669, NASA Contract NAS 8-31960. Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, January 20, 1977, p. 29.
21. Kissinger, L.D., and Verostko, C.E.: "Inherent Ammonia Removal Potential of the Space Shuttle Orbiter Condensing Heat Exchanger." JSC-08797, NASA Lyndon B. Johnson Space Center Crew Systems Division, Houston, TX, August 23, 1976, p. 10.
22. Ibid., p.14.

APPROVAL

COMPUTERIZED ATMOSPHERIC TRACE CONTAMINANT CONTROL SIMULATION FOR MANNED SPACECRAFT

By J.L. Perry

The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.



J.C. BLAIR

Director, Structures and Dynamics Laboratory

☆ U.S. GOVERNMENT PRINTING OFFICE 1993-733-050/80086